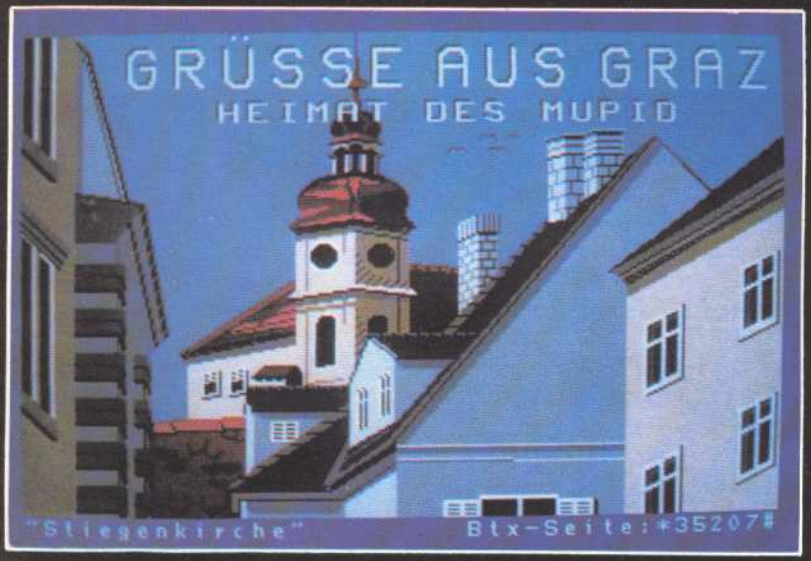
Ára: 30 Ft

mikro számítógép magazin

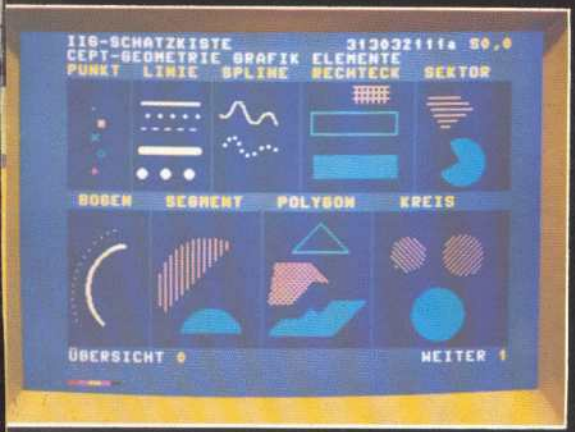
...VAGY AMIT AKARTOK!

1987/10



. . . Ki ne szeretné, ha kazettán vagy hajlékonylemezen garmadával sorakozna könyvespolcain több ezer oldalnyi friss érvényű lexikális és napi aktualitású információ, esetleg igény szerinti felhasználói program? „Ha van MUPID-od, ez a vágyad teljesül."

(Cikkünk a 28. oldalon.)



5. ÉVFOLYAM
1987/10. SZÁM

μ mikro számítógép magazin

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

A kiadvány a Tudományszervezési és Informatikai Intézettel együttműködve készül

A szerkesztőbizottság vezetője:
Kovács Győző

E számunkat szerkesztették:

Bakos Tamás
(programozástechnika)

Broczkó Péter
(hírek)

Kovács Győző
(levelezés)

Lindner László
(sakkprogramozás)

Petróczy Judit
(könyvek)

Simonyi Endre
(klub)

Varga András
(iskola-számítógép)

Címképünk:
Ramocsai Imri munkája

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levélcím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat

Felelős kiadó:
dr. Petrus György
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlapelőfizetési és Lapellátási Irodáján
(1900 Budapest V.,
József nádor tér 1.)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti
a Kultúra,
1389 Budapest, pf. 149.
és a Magyar Média
1932 Budapest, pf. 279.
86—0253

Szikra Lapnyomda
Budapest (87-1321)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Tartalom

A TIT és a tanárok 2
OKTA-TOTÓ 16
Mit tud a SYMPHONY? 22
Segít a vonalkód 26
Móra és a számítógép 27
Miért nem stupid a MUPID? 28
Az IBM megint robbant 31
Szemüveg alatt a számítástechnika ártalmai 32
Már nem bunda a bunda 34
Adok—veszek—cserélek 38
μINFORM 39
Olvastunk . . . 40

ISKOLA-SZÁMÍTÓGÉP

Oktatók változó szerepben 3
UNICOMP vagy TechnoMIR? 6
A MESTOPRI titka 7
Nemzetközi számítástechnikai programozási verseny Bulgáriában 11

DIÁKROVAT

Magyar nyelvű BASIC C64-re 12
Spectrumra írt bináris átalakító 13
Spectrum reflex 13
Mastermind C64-re 14
Több szólam Homelabra 15

PROGRAMOZÁSTECHNIKA

BASIC és gépi kód 17
Z80 programok haladóknak Spectrumra és Primóra 17
Formázott listázás 21

μPROGRAMOK

Korszerűbb 33
Beta BASIC 33

μKLUB

Integrált szoftver 36
Két botkormánnyal 38

SAKKPROGRAMOZÁS

A játékfa és kiértékelése 42
Gépi ellenfeleink 43

AZ OLVASÓ ÍRJA 46

KÖNYVEK 47

HÍREK, ÉRDEKESSÉGEK 48

„Gondolkodás nélkül tanulni haszontalan; Tanulás nélkül gondolkodni veszélyes."
(Konfucius)

A TIT és a tanárok

Az idén nyáron különleges élményben volt részem: három alkalommal közel háromszáz közép- és általános iskolai tanárral találkoztam és beszélgettem a Tudományos Ismeretterjesztő Társulat budapesti szervezetének jóvoltából. Hetedik éve rendezik ugyanis közép- és általános iskolai tanárok részére a szünidei továbbképzést, amely iránt az érdeklődés — magam is így láttam — egyre inkább növekszik.

Sztrókay Kálmánnal, a tanfolyamok „spiritus rector"-ával beszélgettem az eddigi találkozók tapasztalatairól, ezutáni terveikről. Elmondta, hogy a tanfolyamra beiratkozott hallgatók nagyobbik része először vesz részt ezen a továbbképzésen, és az idén úgy látszik, hogy az általános iskolai tanárok érdeklődése megelőzte a középiskolai tanárokét.

A helyzet teljesen világos, hiszen az általános iskolák nagyobbik része nemrégen kapott számítógépeket — ma az általános iskolák cca 80%-a gépesített —, és ezért az érdeklődő pedagógus minden alkalmat megragad, hogy ezt a számára új, de a napi munkájában jól használható berendezést megismerje. A középiskolákban már évek óta vannak számítógépek, így aki meg akarta tanulni a gépek használatát, túlvan rajta. A hallgatók egy része — az újak — ezért elsősorban a kezdő BASIC-re jelentkeztek, míg akik már többször is részt vettek ezeken a TIT-iskolákon, a haladó BASIC-et, de nagyon sokan az ASSEMBLER kurzusokat látogatták. Az előadásokat Poronyi Gábor, a Baranya Megyei Pedagógiai Intézet munkatársa vezette.

Az idei tanfolyamok közül kettőnek a házigazdája a hódmezővásárhelyi Kossuth Szakközépiskola volt, ahol Hajdu János nem csak a számítógépes vezérlés oktatását vállalta, de segített a tanuláshoz a megfelelő szakmai — gépi — környezetet is biztosítani. Egy tanfolyam volt Fonyódon, a Karikás Frigyes Szakközépiskolában, itt Horváth István, a kollégium igazgatója teremtett a számítógépes gyakorlatokhoz igen ideális körülményeket.

Az ember el sem hinné, hogy az ötnapos középiskolás, illetve a négynapos általános iskolás tanfolyami programba mennyi minden belefért; igaz, a foglalkozások reggeltől sokszor késő estig folytak. Magam is tanúja voltam egy a hallgatók és a tanfolyam vezetője közötti „nézeteltérésének", nem akarták ugyanis elfogadni, hogy a programban múzeumlátogatás is szerepelt, amikor a magas szintű nyelvekről szóló előadásokat is folytathatták volna. A vita — félreértés ne essék — nem a hallgatóság kultúra iránti érzéketlenségét jellemezte, inkább azt, hogy az idő minden percét új számítástechnikai ismeretek megszerzésére szerették volna fordítani.

Így ezután már érthető, hogyan fért bele egy-egy szűk hétbe a magas szintű nyelvek előadásai és gyakorlatai mellé Nyíredi Lászlónak, a székesfehérvári Ybl Miklós Szakközépiskola tanárának technikai aspektusú előadása a KFKI-ban kifejlesztett logikai tábláról, valamint Theisz Györgynek, a székesfehérvári József Attila Gimnázium tanárának a számítástechnikai oktatás tapasztalatairól szóló ismertetése, amelyeket a hallgatóság igen szívesen fogadott. Bevált „receptje" ugyanis ezeknek a TIT-tanfolyamoknak, hogy az előadók zöme maga is gyakorló tanár, így pontosan ismeri azokat a problémákat, amelyek a hallgatókat foglalkoztatják; a napi gyakorlatból tud válaszolni a feltett kérdésekre.

A tanfolyam rendezőinek érdeme, hogy a hallgatók nem csupán számítástechnikai előadásokat hallgathatnak, de hozzászólhatnak az iskolák számítástechnikai szervezési, oktatási, számítógép-ellátási kérdéseihez is. Az idén is nagy érdeklődéssel kísért előadást tartott Páris György, az iskola-számítógép programért felelős Tudományszervezési és Informatikai Intézet igazgatója, illetve a Videoton TVC számítógépének iskolai alkalmazásáról Varga András, aki ugyancsak az intézet munkatársa.

Én a tanfolyamon két kedvenc témámról is beszélhettem, az informatika történetéről, illetve a távtanulási technológiáról.

Bevallom: nekem mindig jóleső élmény, ha sikerül a mondanivalóm iránt a hallgatóság érdeklődését felkeltenem. Az előadások utáni beszélgetések azt mutatták, hogy mind a technikatörténet, mind pedig a technologizált nyílt oktatás nagyon érdekelte a részt vevő tanárokat, hiszen — főleg az utóbbi — óriási segítséget adhat az iskolai munkában a számítástechnikai ismeretek elmélyítéséhez és gyakorlati alkalmazásához.

Az előadás utáni beszélgetéseken, de a fonyódi kerekasztal-vitán is teljesen világossá vált, hogy a magyar pedagógustársadalom egyre nagyobb része már nem pusztán érdeklődésből nézegeti a számítógépeket, hanem fel is akarja használni a tantermi foglalkozásokon is. Azt is szeretnék, ha a gépek segítenék a diákokat a tanulásban, és így a gépesített tanulás az oktató munkájukat eredményesebbé és könnyebbé is tehetné.

Ma a pedagógusok egyre inkább megértik — mondták —, hogy a számítógép igen fontos eszköz az iskolában, de még fontosabb, hogy a géphez megfelelő, pedagógiailag, módszertanilag is kiváló oktatási programok álljanak a rendelkezésükre, amelyeket nem programozók, hanem szaktanárok készítettek. Azt hiszem, ezért fogadták szívesen az oktatóprogramok készítését segítő szerzői rendszerekről (authoring systems) és általában a számítógéppel segített oktatás (CAI Computer Aided Instruction — CBT Computer Based Training) technológiájáról szóló beszámolót, azt is remélve, hogy ezek a keretrendszerek hamarosan nálunk is elérhetők lesznek. Hasonló érdeklődéssel fogadták a közönség — így az iskolák — részére is elérhető nyilvános adathálózat, a VIDEOTEX ismertetését is; a témával kapcsolatban feltett kérdésekből világosan látszott, hogy ezt a lehetőséget — ha kézzelfogható lesz — az iskolák nagyon gyorsan alkalmaznák.

Beszélgetéseim a tanfolyamon részt vevő tanárokkal sajnos arról is meggyőztek, hogy az oktatástechnológia eredményei ma még nehezen jutnak el főleg a vidéki tanárokhoz, pedig egy számítógépekkel támogatott megújulás nélkül nehezen lehet az oktatás hatékonyságát a jövőben tovább növelni.

Ezért tartom én a TIT kezdeményezését hasznosnak, ha úgy tetszik hézagpótlónak; egy olyan továbbképzési akciónak, ami a magyar oktatásügy „nagy kalandját", az iskola-számítógép programot eredménnyel támogatja.

KOVÁCS GYŐZŐ

Oktatók — változó szerepben

Egy felmérés tanulságai

A magyarországi iskolaszámítógép-program híreiről, eredményeiről, örömeiről és gondjairól ugyan rendszeresen kapnak tájékoztatást a Mikroszámítógép Magazin olvasói, mégsem érdektelen a dolgokat időről időre összefoglalva, másképp is áttekinteni.

Az 1983 szeptemberében indult programnak három alapvető iránya volt:

- A megfelelő számú gép szétosztása és az üzemeltetés, szerviz megszervezése
- Az iskolákban meglévő gépekhez jó színvonalú szoftverek tervezése, kifejlesztése, előállítása és terjesztése
- A pedagógusok alapképzése és továbbképzése

Az elmúlt négy iskolaév során elsősorban a számítógépek típusainak kiválasztása, a hardver vásárlása volt az iskolák érdeklődésének homlokterében. A legutóbbi időben figyelhettünk fel a szoftver iránti fokozódó igényekre. Ritkábban esett és esik szó a tanárok alap- és továbbképzéséről. Most erre kerítünk sort.

1984-ben és 1985-ben az Országos Oktatástechnikai Központ kétéves pedagógus-továbbképzési programot dolgozott ki, melynek eredményeként létrejött a megyei szakreferensek hálózata, és több ezer középiskolai tanár kapott számítástechnikai alap- és továbbképzést. Ez természetesen nem jelentette azt, hogy azok a tanárok, akik részt vettek a továbbképzésen, ettől kezdve mindennapi munkájukban rendszeresen alkamazták is a mikroszámítógépet. Nézzük mindenesetre az előzményeket!

Az iskolaszámítógép-program indulásakor, 1983 szeptemberében országszerte 1200 középiskolai tanár vett részt felkészítő tanfolyamon. Ezután meglehetősen gyorsan kezdtük meg a tapasztalatok gyűjtését és feldolgozását. 1983 decemberében 24 kérdésből álló kérdőívet küldtünk ki 1200 tanárnak, hogy képet nyerjünk az alkalmazás kezdeteiről. A kérdőíves felmérésben részt vett tanárok körében 1985-ben attitűdvizsgálatot is végeztünk.

Az Oktatástechnológiai Tárcaközi Tudományos és Koordináló Tanács (OTT) 1986 novemberében 25 újabb kérdést küldött ki a fentiek közül kiválasztott 700 tanárnak. 297 értékelhetően kitöltött kérdőív érkezett vissza, melyek feldolgozásának néhány érdekes eredményét az alábbiakban mutatjuk be.

A vizsgálatban résztvevők összesített adatai:

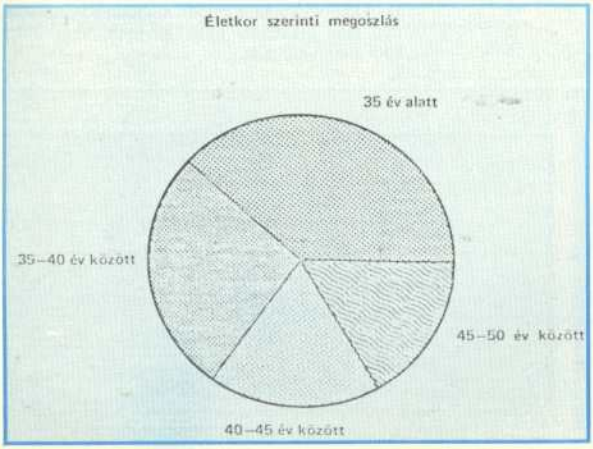
Nők	
Középiskolai tanár	71
Általános iskolai tanár	10
Férfiak	
Középiskolai tanár	207
Általános iskolai tanár	9

A válaszadók életkor szerinti megoszlása:

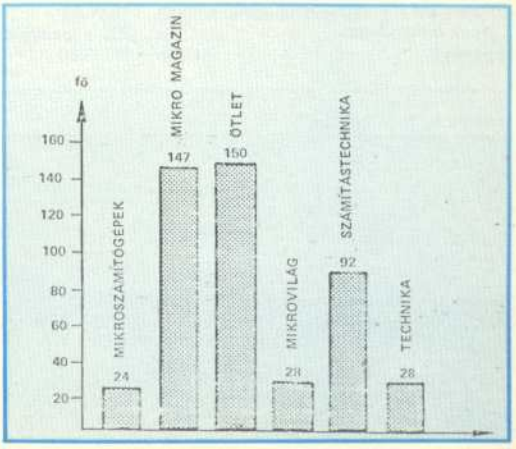
35 év alatti	116
25—40 év közötti	81
40—45 év közötti	56
45—50 év közötti	16
50 év feletti	28

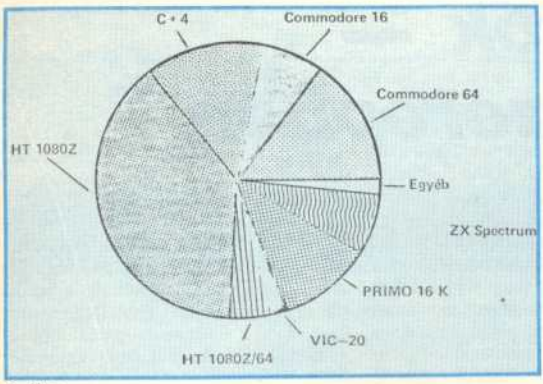
Az életkori adatok tükrözik, hogy az iskolák elsősorban a 40 év alatti tanárokat jelölték ki a számítógépek iskolai bevezetésére *(1. ábra)*. Ez annál is inkább helyes döntés volt, mert a 35 évnél fiatalabb tanárok közül a természettudományi és műszaki te-

1. ábra

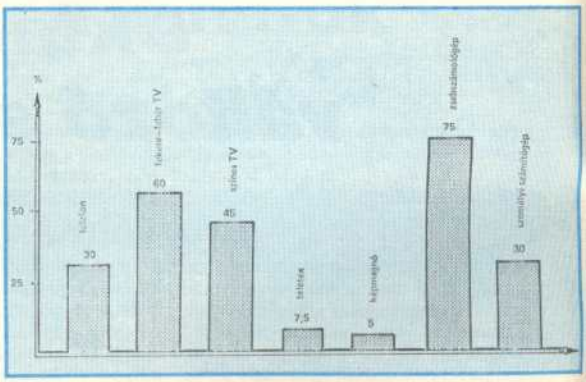


2. ábra





3. ábra



5. ábra

rületeken végzettek közül 78-an már képzésük során is — igaz, eltérő mértékben — megismerkedhettek a számítástechnika alapjaival.

A válaszadók többsége az alapképzések, a tanfolyamok mellett rendszeresen figyelemmel kíséri a magyar nyelvű szakirodalmat. Sajnálatos tény, hogy tanáraink csak a magyar nyelvű könyveket és folyóiratokat olvassák rendszeresen. A jelzések szerint ennek nem a nyelvtudás hiánya a fő oka, hanem hogy az idegen nyelvű folyóiratokat — elsősorban vidéken — nehéz megszerezni, kölcsönözni.

Arra, hogy: „Sorolja fel azokat az információforrásokat, sajtótermékeket, kiadványokat, amelyek segítséget nyújtanak pedagógiai, oktatástechnikai tevékenységében" a *2. ábra* szerinti válaszokat kaptuk. Tehát a folyóiratok közül legnagyobb a Mikromagazin és az Ötlet/Bitlet olvasottsága. A kiadványok hasznosulási rangsorából az OOK által kiadott Mikroszámítógépek példatársorozatot emeltük ki. A „Milyen típusú számítógépe(ke)t használ az iskolában?" kérdésre küldött válaszok *(3. ábra)* alapján a gépek mennyisége:

HT 1080Z	818 db
C64	309 db
C +4	286 db
Primo 16 k	220 db
ZX—Spectrum	150 db
C16	144 db
HT 1080Z/64	102 db
VIC—20	35 db
Egyéb	28 db

Ismereteink szerint ez megfelelt a felmérés időpontjában a középiskolák magyarországi általános helyzetképének. A HT 1080Z gépek túlsúlya — mint ismeretes — abból adódik, hogy az 1983/85-ös tanévben központi forrásokból ezeket a gépeket kapták meg a középiskolák.

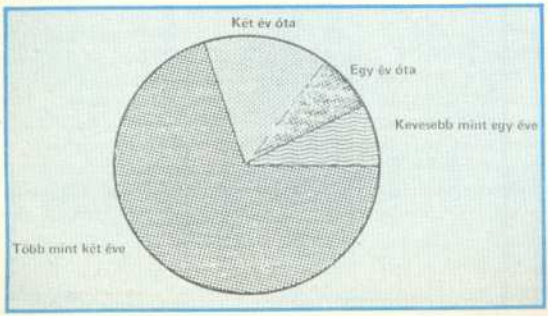
A kérdőív „Milyen típusú személyi számítógépe(ke)t szeretne használni a jelenlegiek helyett (mellett)?" kérdésére az alábbi válaszok érkeztek:

C64	85
C +4	65
Primo 64 k	33
TV-Computer	26
IBM AT	21
ZX—Spectrum	13
HT 3080	10

A felmérés kiterjedt arra is, hogy: „Mióta használja rendszeresen tanítványaival a számítógépet?" A *4. ábráról* látható, hogy a válaszoló tanárok többsége már több mint két éve alkalmazza a számítógépet. Egy másik kérdés segítségével arra kaptunk adatokat, hogy a tanárok a korszerű informatikai eszközök közül melyekkel rendelkeznek otthonukban. Az összesített adatokat az *5. ábra* mutatja. Eszerint mindenkinek van tévéje, néhányuknak színes és fekete-fehér készüléke is. Az országos átlagnál jobb a telefon- és a személyiszámítógép-ellátottság. Kiemelkedő a zsebszámológépek magas aránya; ez azonban érthető, hiszen számítástechnikával rendszeresen foglalkozó tanárokról van szó.

Az egyik kérdés az írásvetítő, a diavetítő, a filmvetítő, a képmagnetofon, a központi tévéműsor és a személyi számítógép felhasználásának gyakoriságára kérdezett rá *(táblázat)*. Az írásvetítőt a tanárok kedvvel és gyakran alkalmazzák; a megkérdezettek közel kétharmada naponta vagy hetente használja. A diavetítőknél már kevésbé kedvező a kép, hiszen a válaszadó tanárok fele ritkábban mint havonta, vagy soha nem vetít diát.

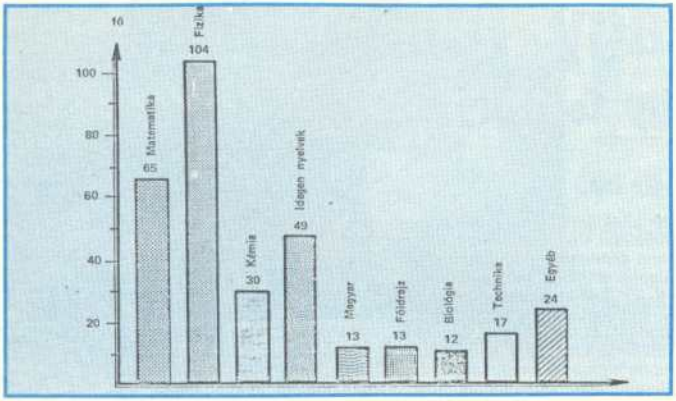
4. ábra



MILYEN GYAKRAN HASZNÁLJA ?

N = 297

ESZKÖZ	NAPONTA	HETENTE	HAVONTA	RITKÁBBAN	SOHA
ÍRÁSVETÍTŐ	37	93	53	46	7
DIAVETÍTŐ	14	41	53	120	38
FILMVETÍTŐ	8	13	48	140	50
KÉPMAGNETOFON	4	47	64	88	56
KÖZPONTI TV-MŰSOR	8	8	14	73	103
SZEMÉLYI SZÁMÍTÓGÉP	116	113	57	25	—



6. ábra

A filmvetítést tekintve kimondottan rossz a helyzet: a megkérdezettek kétharmada ritkábban, mint havonta, vagy sohasem vetít filmet. A film közismerten a legdrágább taneszköz, a fővárosi és a megyei filmgyárakban ezerszámra találhatók a 16 mm-es oktatófilmek, de sajnos az iskolák ezeket csak ritkán kölcsönzik.

Világszerte megfigyelhető, hogy a video kiszorítja a filmet az oktatásból. Sajnos, nálunk még nem erről van szó. A videózásbeli eredmények még igencsak szerények.

Az eddig csekély hatékonyságú központi tévéműsorok és az ITV szerepe is változni fog. A video fokozatos térnyerésével a központi adások növekvő százalékát fogják rögzíteni az iskolák, és ily módon az adások hasznosulása is remélhetőleg növekedik.

A személyi számítógép alkalmazási gyakoriságát firtatta a következő kérdés. Ezzel kapcsolatban fel kell hívnunk arra a figyelmet, hogy ezek az adatok bizonyára lényegesen kedvezőbbek az országos átlagnál, hiszen a válaszolók éppen a számítástechnikában élenjáró tanárok.

A kérdés így szólt: „Mely témakörökben és tantárgyakban tudta eddigi munkájában leghatékonyabban alkalmazni a mikroszámítógépet?"

Kiemelkedő volt a matematika (194) és a fizika (182) döntő többsége, a harmadik legmagasabb érték a technika (35) volt. Röviden néhány példa arra, hogy a fenti tantárgyakban mely témakörök voltak a legygyakoribbak.

Matematika

Függvények	41
Kerület-, területszámítás	39
Valószínűségszámítás	19
Egyenletek	13
Sorozatok	41

Fizika

Hullámjelenségek	25
Pontszerű testek mechanikája	22
Atomfizika	15
Merev testek mechanikája	13
Halmazállapotok és hőjelenségek	10
Elektromosságtan	9

Technika

Számítástechnika	10
Mérésszabályozás	6
Műszaki rajz és ábrázolás	5

Néhány kérdés a jövőt tudakolta. Például: „Az ön véleménye szerint a soron következő időszakban mely témakörökben, tantárgyakban lehet leghatékonyabban alkalmazni a mikroszámítógépet?"

A *6. ábra* mutatja, hogy mely tantárgyak iránt nyilvánul meg a legnagyobb érdeklődés. Felsorolunk néhány gyakori témakört az egyes tantárgyakon belül.

Fizika: mozgások, mérések, statisztikus fizika, kísérletek modellezése és kiértékelése, szimuláció, hullámtan.

Matematika: egyenletek, geometria, függvények, számelmélet, valószínűségszámítás.

Kémia: szimuláció, számonkérés, mérések.

Idegen nyelv: gyakoroltató, szavak tanulása, nyelvtani fogalmak alkalmazása.

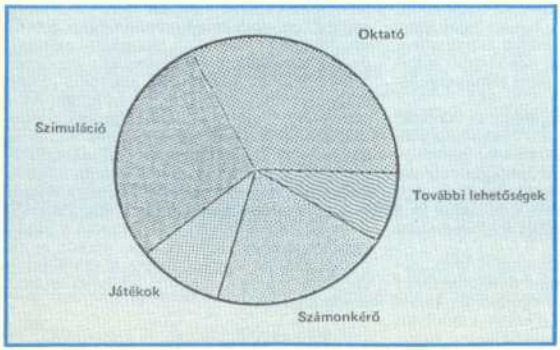
A szoftverek helyzetéről kapott kép is érdekes: a *7. ábra* mutatja az arányokat. Jelenleg a szimulációs és az oktató szoftverek vezetnek. De a „Véleménye szerint milyen típusú szoftverre van leginkább szükség az elkövetkező időszakban?" kérdésre is választ kértünk, így az elvárásokat a *8. ábra* tükrözi. Láthatóan továbbra is az oktató és a szimulációs szoftverek iránti kereslet lesz a legnagyobb.

A pedagógusok szerepének döntő fontossága e felmérés adataiból is világosan kitűnik. Az eredmények fokozásához tehát a tanárok irányító/vezető szerepét a számítógépes oktatási alkalmazásoknál nem lehet eléggé erősíteni.

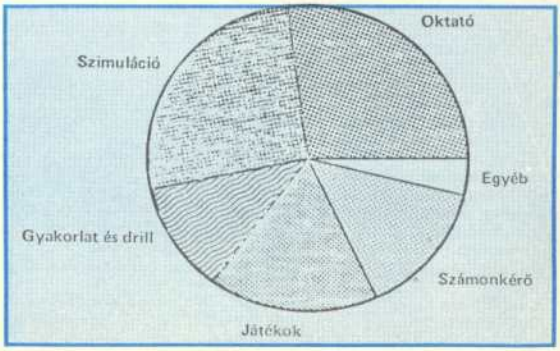
SZŰCS PÁL

A rovatvezető megjegyzése: Ma már az általános iskolákban is megindult a program. Bár iskolánként kevesebb, de abszolúte több gép van az általános iskolákban, mint a középfokú oktatási intézményekben. Az általános iskolai pedagógusok képzése jóval nagyobb és nehezebb feladat a középiskolaiaknál . . .

7. ábra



8. ábra



UNICOMP vagy TechnoMIR?

Megjegyzések egy tanfolyam margójára

Még a tavaszi szünetben továbbképzést szervezett a Tudományszervezési és Informatikai Intézet általános és középiskolákban számítástechnikával foglalkozó tanárok részére, amelyen a számítógép és környezetének kapcsolatával foglalkoztak. Az ország valamennyi középiskolájába, a jelenlegi elektronizációs program keretében pedig mostanáig az általános iskolák jelentős részébe is eljutottak a személyi számítógépek, amelyek az első, rácsodálkozó, tapogatódzó lépések után folyamatosan beépülnek az iskolák oktató-nevelő munkájába. Egyre több tanár vállalkozik a „nagy kalandra". Arra, hogy óráján vagy az iskolai élet más területén használja az iskola számítógépeit. A középiskolák túlnyomó többségében azonban a számítógépet csak az ismétlődő számítási feladatok elvégzésére, jobb esetben egy-egy tantárgy anyagrészéhez kapcsolódó jelenség demonstrálására használják.

A számítógépet mint egy információs rendszer elemét talán még kevesebben alkalmazzák, s ez is általában kimerül az osztály- vagy iskolaátlagot számoló programokkal. Csak elenyésző számban áll össze a dolog nyilvántartássá, adattárrá vagy hálózattá. A számítógépes folyamatvezérlés vagy -szabályozás lehetőségét pedig csak néhány, speciális helyzetben levő iskola aknázhatja ki. No nem azért, mert olyan különleges számítógépekkel rendelkeznek, hiszen ehhez a feladathoz nem is a számítógép típusa, hanem sokkal inkább a géphez csatlakozó periféria a döntő, amely a környezettel való kapcsolatot biztosítja.

Az iskolák között a különbséget a számítógépek felhasználása szempontjából valójában a rendelkezésre álló perifériák skálája befolyásolja, no meg a gépek száma. Kevésbé pedig a manapság oly gyakori viták középpontja, vagyis az, hogy melyik számítógéptípus a legmegfelelőbb a ma iskolájában. Tény, hogy bármely típusról is legyen szó, szinte mindegyiknél hasonló eszközök állnak rendelkezésre a környezettel való kapcsolattartáshoz.

A tanfolyam lényegét éppen ez adta. Mit tegyen egy felhasználó, ha a külvilág információit gyorsan, pontosan, lehetőleg hiba nélkül kívánja feldolgoztatni? A válasz egyszerű: olyan univerzálisan használható perifériát (interfész+érzékelő, illetve interfész+beavatkozó) csatlakoztasson a számítógépéhez, amely a környezetből az információt begyűjteni, illetve amely a külvilág folyamataiba beavatkozni képes. Ha lehet, ezt olcsón, könnyen kezelhetően, egyszerűen tegye.

A tanfolyam keretében két interfész családdal ismerkedhettünk meg. A miskolci Zalka Máté Gépipari Szakközépiskola szakemberei által kifejlesztett UNICOM 3M nevű interfészt C64-hez csatlakoztatva mutatták be a fejlesztők. Az eszköz több, egyetlen készülékvázba elhelyezett áramköri modulból áll. Többek között tartalmaz A/D, D/A, frekvenciamérő, motorvezérlő áramköri modulokat. Az iskolai felhasználás szempontjából különösen előnyösnek tűnnek a modulok kedvező műszaki paraméterei, valamint megbízhatóságuk, amely a konstrukcióból következik. Ez főleg a szakközépiskolák számára érték, hiszen az áramkörök pontos működése alkalmassá teszi a rendszert a számítástechnika ipari felhasználásának bemutatására. Ilyen jellegű felhasználásra példa a CNC vezérlésű esztergagép programozása során a számítógép és az eszterga közötti illesztés. Sokat sajnos nem tudunk erről a rendszerről, de talán később részletesen megismerhetjük. Egy hátrányos tulajdonsága azonban mindjárt szembeötlik: az interfészmodulok egy közös készülékben helyezkednek el, így egyszerre kell az egészet megvásárolni, ami a legtöbb iskola számára meggondolandó lépés.

A másik megismert interfészrendszer a TechnoMIR, amelyet az Országos Pedagógiai Intézet megbízásából az ELTE Általános technika tanszék munkatársai fejlesztettek és szabadalmaztattak. Jelenleg az Elektroscan Kisszövetkezet gyártja és a Tudományszervezési és Informatikai Intézet értékesíti a rendszert.

Az interfészrendszer elemei itt is egy-egy funkciót látnak el, de a fizikai megvalósítás olyan, hogy az egységesített méretű kis piros dobozkák egymáshoz tetszés szerinti számban és sorrendben csatlakoztathatók. Az elemek egy közös gerinchálózaton kapcsolódnak egymáshoz és a számítógép buszrendszeréhez.

Az iskolák szempontjából a TechnoMIR rendszer első előnye, hogy a modularitásából adódóan nem szükséges az egész rendszert egyben beszerezni, hanem csak egy-egy elemét, és később tovább bővíthető a készlet. A teljes eszköztár ára viszont indokolatlanul magas, a központi támogatás ellenére is.

A másik nagy vonzerő, hogy az iskolákban leginkább előforduló számítógépek mindegyikéhez csatlakoztatható, és könnyen, BASIC-ből kezelhető, függetlenül attól, hogy Z80 vagy 68xx alapú a gép. A rendszer ezt a problémát a modulok és a gép közötti csatlakozókábel kialakításával oldja meg.

Bár az eszköz meglepően masszív és mechanikailag is stabil, iskolai alkalmazása esetén a tájékoztatóban megadott ki- és bemeneti áramértékeket érdemes komolyan venni. Viszont mind az eszköz, mind a számítógép károsodás nélkül kibírja, ha működés közben szétszedjük az egymásba kapcsolódó modulokat vagy ha újabb modult csatlakoztatunk a gerinchálózatra. Bár a rendszer nem laboratóriumi pontosságú, az iskolák demonstrációs céljaira kiválóan alkalmas. Ezt a megállapítást tapasztalataink alapján merjük terjeszteni, hiszen a tanfolyam nagyrészt a modulok alkalmazásának egyéni gyakorlásából állt. Sőt olyan modulokat is kipróbálhattunk, amelyek még fejlesztés alatt voltak. Ilyen az a modul, amellyel a számítógépeket hálózatba lehet kötni, vagy ilyen a léptetőmotort vezérlő modul is.

A Tudományszervezési és Informatikai Intézet jóvoltából a tanfolyamon részt vevő szinte valamenyi tanár külön számítógépen és a hozzá csatlakozó TechnoMIR elemeken gyakorolhatta a rendszer használatát.

Bár a tanfolyam szervezői az interfészrendszer dokumentációját nem biztosították, mégis a modulokkal való rövid ismerkedés után sok érdekes felhasználás született. Néhány soros BASIC program segítségével motorok fordulatszámát mérhettük, forgalomirányító lámpák vezérlését szimulálhattuk LED-ekkel, vagy — sokak tetszését ez nyerte el leginkább — számítógépek közötti adatátvitelt valósíthattunk meg üvegrúd, LED és fotodióda felhasználásával.

A modulok számtalan egyéb felhasználási lehetőségét próbálhattuk ki az egyéni fantáziának és ötleteknek megfelelően. Így született tűzriasztó és tűzoltó rendszert szimuláló program, amely termisztort figyelt, és Trabant ablakmosó szivattyút vezérelve oltotta el a gyufa lángját. Említhetjük a fémépítő játékból készült liftet vezérlő programot vagy az elektronikával foglalkozók figyelmébe ajánlható, néhány TTL IC logikai tesztelését végző összeállítást is.

Rövid tájékoztatónkba nem fér az egyes modulok részletes ismertetése és felhasználási tanácsok közlése, ezért egy külön sorozatban foglaljuk majd össze mindazt a tudnivalót, amit a modulok használata előtt érdemes elsajátítani. A sorozattal a rendszer felhasználását szeretnénk segíteni, minél több kipróbált, a gyakorlatban bevált összeállítás bemutatásával. Természetesen várjuk mindazok ötleteit, akiknek van már valamilyen TechnoMIR moduljuk, és szívesen közreadnák tapasztalataikat.

Azok leveleit is várjuk, akiknek valamilyen más, akár saját készítésű, akár vásárolt és az iskolában bevált interfész van a birtokukban. Kérjük, küldjék el az eszköz és az összeállítások leírását, képét a szerkesztőségbe, az Iskola—Számítógép rovat vezetőjének nevére.

KIRÁLY LÁSZLÓ—ALBU LÁSZLÓ

Oroszóra számítógéppel

A MESTOPRI titka

Nem tudom, hogy kinek milyen emléke van a középiskolai oroszórákról, de én bevallom, hogy igen nagy szorongással néztem eléjük. Beismerem saját lustaságomat, de mégsem azon múlt a dolog — valahogy nekem ez nem feküdt. Nem követendő példaként írom, hanem őszintén megmondom akkori érzéseimet. Ma már nagyon sajnálom érdektelenségemet, mert az orosz nyelvtudás néha nagyon hiányzik.

Különös kíváncsisággal fogadtam tehát azt a meghívást, hogy látogassam meg a budapesti Radnóti Miklós Gimnázium első osztályosainak „számítógépes" oroszóráját.

A tanulók már az óra előtt nagy izgalomban voltak; nem láttam rajtuk az ifjúkoromból emlékezetes lógási szándékot. Három-négy gyerek ült egy gép előtt, amelyen oktatóprogram futott. Minden csoport másmás programmal gyakorolt. A szemek csillogása elárulta, hogy igazán élvezik, amit csinálnak.

Tanárnőjük, dr. Fülöpné Kalas Mária elmondta: kolléganője inspirálta, hogy próbálkozzon meg a nyelvoktatásban a számítógéppel. Több mint harmincéves oktatási tapasztalattal rendelkezve, amint őszintén bevallotta, ő is azt hitte, mint a kollégái, hogy majdnem tökéletes didaktikai módszereket alkalmaz. Éppen ezért először maga is megküzdött a kétkedéssel, és a kezdeményezés az iskolában is elég nagy feltűnést keltett. Főleg a reál tárgyakat oktató tanárok hitetlenkedtek. Számára is nagy meglepetés volt az új módszer átütő sikere és eredményessége. Mivel a tanulók különösen igénylik az azonnali kiértékelést, talán annak is köszönhető az újmódi oktatás népszerűsége, hogy a gép ezt megteszi.

A konkrét tapasztalatok az alábbiak.

Nem valami újkeletű mozaikszóval szándékozunk megismertetni olvasóinkat, amikor a MESTOPRI-ról beszélünk. A név egy ZX-Spectrumra, Commodore 64-re és Plus/4-re egyaránt elkészített nyelvi oktatóprogramot takar. Az oroszul tudóknak talán könnyebben adódik a megfejtése, ha eláruljuk, hogy a местоимения (névmások) és a прилагательные (melléknevek) elvonásából származtatjuk: MESTO + PRI.

Mit tud a MESTOPRI?

Jó tanárai voltak, hát megtanult oroszul. Annyira persze még nem, mint akik az anyatejjel együtt szívják magukba az orosz szót, de annyira már igen, hogy mintegy harmincezer mondatot előállítson ezen a nyelven, és ezeket cirill betűkkel meg is jelenítse a képernyőn *(1. ábra)*. Sok vagy kevés a harmincezer mondat? Hosszú idő telik el általában, mire valaki ilyen mélységben birtokol egy nyelvet, és a MESTOPRI-éhoz hasonló biztonsággal kezel a nyelven belül bizonyos témaköröket. Itt ugyanis csupa olyan mondat születik, amelyben ott van egy-egy főnév társaságában az orosz melléknév. Állhat ez egyes vagy többes számban, a függő esetek (tárgy, birtokos, részes, eszköz, elöljárós) bármelyikében, és ami nagyon fontos, hogy megtalálhatjuk a весёлый és a средний, tehát a kemény és a lágy tőtípusokat. Az effajta főnév-melléknév összetételekhez szorosan kapcsolódnak a harmadik személyű он, оно, она, они személyes névmások ragozott alakjai. Bármikor helyettesíthetjük velük a hosszú, megannyi hibázásra lehetőséget adó és a már elhangzott szerkezeteket. Így:

Я жил у киевских друзей. →Я жил у них.

Amint a példánkból is kitűnik, a személyes névmások használatánál ügyelnünk kell az elöljárószókra. Ezek függvényében változik meg egy-egy н-nel az alakjuk *(2. ábra)*.

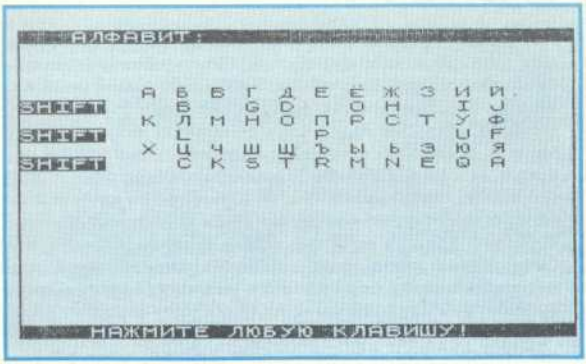
A program felkészült az ilyen formák képzésére, és nem kerülik el a figyelmét a ragozás során szerepet játszó hangtani szabályszerűségek sem (к, г, х stb.).

A MESTOPRI alapvetően két, egymással állandó kapcsolatban levő menürendszerrel dolgozik. Az első — elágazásainak megfelelően — mutatja be a melléknevek és a névmások ragozását. Külön tanulmányozhatjuk a nőnemű, hímnemű és a semlegesnemű, valamint a többes számú formákat *(3., 4., 5. és 6. ábra)*.

E szemléltető táblázatok a második menürendszerben a segítség (HELP) szerepét játsszák. Itt fogalmazódik meg a feladat, hogy nekünk, felhasználóknak kell számot adnunk ragozási ismereteink színvonaláról. Látszólag egyszerűen elvégezhető a véletlen sorrendiséggel — mindig más szövegkörnyezetben — megjelenő mellék-

nevek végződésének kiegészítése, majd a főnév-melléknév szerkezetek névmásokkal való helyettesítése. A helyes megoldáshoz azonban szükség van az összes helyesen ragozott alakra. Ha elvétjük a feleletet, a MESTOPRI — fokozatosan segítve — eljuttat bennünket a jó megoldáshoz. Előbb csak figyelmeztet a hibánkra, majd ajánlja, hogy kérjünk segítséget *(7. ábra)*. Ha az első blokkból már ismerős HELP láttán újra elrontjuk a választ, maga a program jelzi, hogy melyik esetet nem ismertük fel, aztán pedig a mondatban is megjeleníti, amit nem sikerült jól megadnunk.

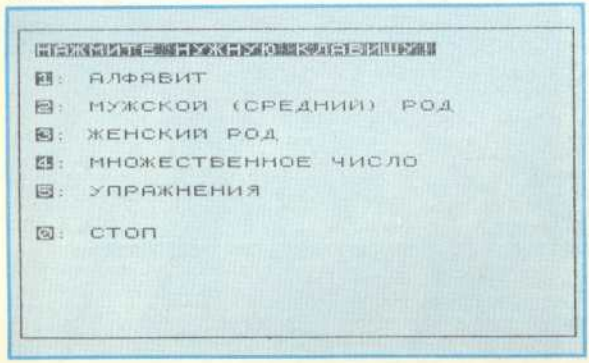
Igen fontos, hogy mi választhatjuk ki, milyen esetből kérünk mondatokat *(8. ábra)*. Módunkban áll önállóan, minden mástól függetlenül gyakorolni például a részes vagy éppen a birtokos ala-

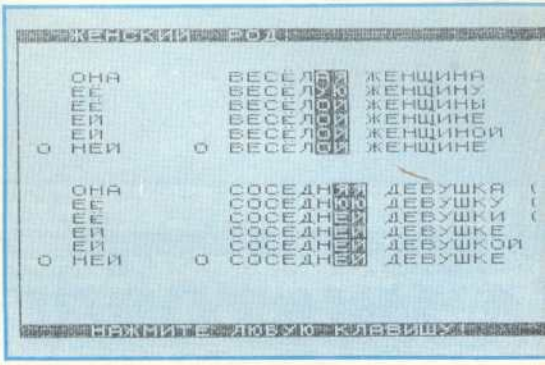


1. ábra

2. ábra

3. ábra

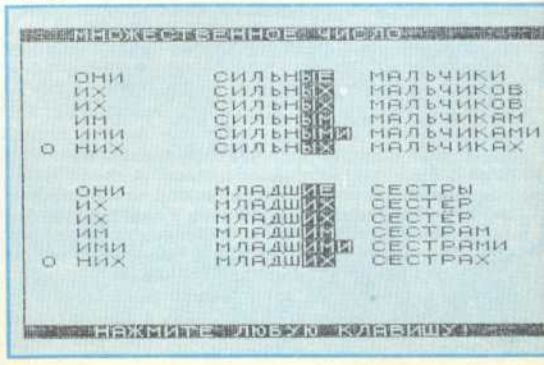




4. ábra

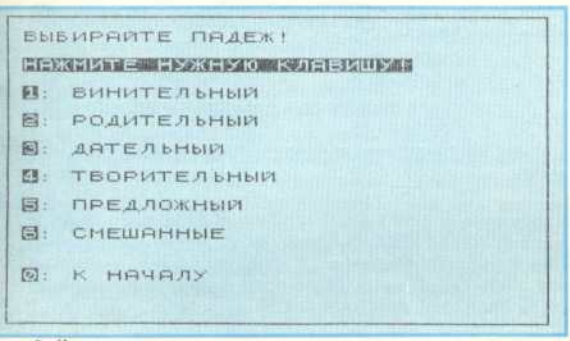
5. ábra

6. ábra



kok képzését. Ezen belül maradva persze már a program dönti el, hogy melyik formát kérdezi. Ha a vegyes típust (6) kérjük, akkor arról is a program határoz, hogy melyik esetből generálja a kérdést. Amikor befejezzük a munkánkat, összesítő értékelést és egyúttal minősítést kapunk *(9. ábra)*, ami további versenyre, küzdelemre ösztönözhet bennünket.

7. ábra



8. ábra

9. ábra

Mi történt a Radnótiban?

A MESTOPRI az elmondottak birtokában vállalja velünk a közös tanulást. Így tette ezt Fülöpné Kalas Mária tanárnő és e cikk szerzőjének irányítása mellett a Radnóti gimnázium I. A osztályában tizenkilenc diákkal.

A középiskolák első osztályaiban kötelező nyelvtani anyag a lágy tövű melléknevek ragozása, és némi átcsoportosítással célszerű itt foglalkozni a névmásokkal is. Ezért dönthettünk úgy, hogy próbára tesszük a MESTOPRI erejét, azaz számítógépes oroszórát szervezünk. Egy ilyen terv kivitelezése ma még — legalábbis orosz nyelvből — úttörő vállalkozásnak számít. Egyrészt azért, mert a tanulók többségéről még mindig hajlamosak vagyunk — főleg mi, humán szakosok — azt feltételezni, hogy nem ért a számítógéphez. Másrészt azért, mert az órához bizonyos technikai minimumfeltételek szükségesek. A Radnóti gimnázium számítástechnikai laborjában a személyi számítógépeknek olyan választékával találkozhatunk, amely történeti hitelességgel bizonyítja a honi iskolaszámítógép-program megannyi állomását. Ez nem lenne baj, hiszen programozásra bármelyik gép alkalmas. Annyi azonban bizonyos, hogy a 48 k-s Spectrumra vagy a 64 k-s Commodore-ra megírt orosz program nem fut sem a 16 k-s HT-ken, sem a Primókon, de még a 16 k-s Commodore-okon sem.

Szerencsénkre mindkét problémánk megoldódott. Az első kételyt maguk a tanulók oszlatták el. A másodikban pedig segítettek (az OPI, az ELTE stb.). Öt gépet átlagosan minden órára sikerült előkészítenünk, ami távolról sem megnyugtató a jövőt illetően. Az eredményeket és a lelkesedést látva pedig érdemes lenne valamit tenni az egységes géppark kialakításáért — akár iskolák közötti gépcserékkel is. Főként, mert a tanárok és a tanulók részéről egyaránt megvan az igény, a nyitottság, amelynek többek között a szóban forgó oroszórákat köszönhettük.

Orosztanulásra is jó

A témának öt órát szenteltünk. Hármat az anyag tényleges feldolgozásakor tartottunk meg, kettőt pedig három hónappal később, amikor arra voltunk kíváncsiak, hogy számítógéppel vagy anélkül mennyire mélyen rögződött a ragozás.

Első lépésként felmértük, hogy mit tud a majdani számítógépes kollektíva (A csoport), és egy másik, a témát vele párhuzamosan tanuló első osztály (B csoport). A feladat mindössze annyi volt, hogy 10 × 2 mondatban ki kellett egészíteni a melléknévi végződéseket, majd a jelzős főneveket helyettesíteni a megfelelő személyes névmással. Íme ízelítőül néhány típusmondat az első tesztből:

Мы приглашаем соседн . . . девушек. Meghívjuk a szomszéd lányokat. Познакомьтесь с прекрасн . . . сестрой Ирины! Ismerkedjetek meg Irina gyönyörű nővérével! Витя мечтает о нов . . . подарках. Vitya ábrándozik az új ajándékokról.

E teszt eredményei többeknél tanúskodtak komoly hiányosságokról. No, nem az új anyagot, hanem a már régen tanultakat illetően. A tesztet az A csoport mindössze 23,3 százalékra oldotta meg, ellentétben a B csoport nagyszerű eredményével, amely ekkor 73,1 százalék volt. A következő órákon a B csoportban hagyományos módszerekkel — alapvetően frontális munka keretében — folyt a tanítás. Ezalatt az A csoport tanulói megismerték a program működését, kezelését. Négyen jutottak egy-egy számítógéphez. Ezekben az önkéntes társulással kialakított műhelyekben folyt a lényegi munka. Mindenki kapott külön feladatot is azon túl, hogy közösen, egymást nemegyszer mókásan felülbírálva megoldották a mondatokat. Az órák végén, eleget téve a megfigyelési

szempontoknak, sokszor húsz-harminc mondat sorakozott a füzetekben.

Megfigyelési szempont lehetett a nemek szerinti csoportosítás, az egyes és a többes számú mondatok, a kemény és a lágy tövű melléknevet tartalmazó mondatok feljegyzése. Ha munka közben olyan probléma adódott, amelyet a csoport nem tudott megoldani, segítségül hívhatta tanárát, aki egyébként hívás nélkül is bele-belehallgatott a beszélgetésekbe. Magyarázatokkal segítette, irányította a tanulókat, miközben neki magának is volt a Spectrumok és a Commodore-ok „személyében" öt másik segítője.

A harmadik órán volt az összefoglalás, itt nyílt mód a tapasztalatcserére. Mindenki kérdezhetett mindenkit, beszámolhatott a neki problémás feladatokról. Az ezután megíratott újabb teszt összességében ugrásszerű fejlődést mutatott, annak ellenére, hogy ebben már szerepelt magyarról oroszra való fordítás is, tehát a feladat lényegesen nehezebb lett. Ám az A csoport eredménye 73,3 százalékra javult. Igaz, szépen dolgozott a B csoport is: 79,4 százalékot ért el, amit szintén nem tartunk kevésnek.

Meg kell azonban jegyeznünk, hogy ez utóbbi csoportban változatlanul találkoztunk nagyon gyenge, elégtelen szintű teljesítményekkel. Sok olyan hiba ismétlődött, amelyet már az első alkalommal kiszűrtünk. Az A csoport munkája hangsúlyozottan egyenletesebb képet mutatott. Láthatóan megszűntek bizonyos típushibák. A gyerekek felszabadultak, és később, függetlenül a feldolgozott témától, a kezdetekhez képest lényegesen javult munkájuk színvonala.

Három hónap elteltével izgalommal vártuk a harmadik teszt eredményeit. Ezt ismét minden előkészítés nélkül írták meg a tanulók. Az oroszra fordítandó mondatok között most még nehezebbek szerepeltek: *Tizennégy éves az ifjabb Ivanov. Érdekel bennünket az orosz népművészet. Láttad a szomszéd lányt?* Nem volt könnyű egyiket sem lefordítani. Az A csoport átlaga 49,9, a B csoporté 68,7 százalék lett. Ekkor újabb két órát szántunk gyakorlásra. Munkamódszerünk megegyezett a korábbival.

A negyedik teszt, amely összetételében az előzőhöz hasonlított, a következőket mérte: az A csoport 62,2 százalékra, a B csoport 62,5%-ra teljesített.

A *miért*ekre valószínűleg nem lenne célszerű most megadnunk a választ. Azért sem, mert a következő tanévben folytatni szeretnénk a kísérletet, és azért sem, mert sok mindenre még nem tudnánk teljes bizonyossággal felelni. Érdemes azonban elmondanunk, hogy

a MESTOPRI kínálta gyakorlási mód olyan irányba vezet bennünket, ahol nem a merev drillekkel kötünk ismeretséget. Könnyedén, megannyi új és újabb változatban találkozunk a problémával, amelyet megoldani igazán csak az értelem logikájával lehet.

Érdemesnek tartjuk továbbá közreadnunk azt a felmérést, melyet Fülöpné tanárnő készített az I. A osztályosokkal. A csoportból tizenheten válaszoltak. Íme a kérdések és a rájuk adott feleletek:

1. Volt-e már eddig is ilyen közeli találkozásod a kisszámítógéppel? IGEN = 15 fő NEM = 2 fő

2. Van-e otthon a családban kisszámítógép?
VAN = 5 fő NINCS = 12 fő

3. Érdekesebb-e a nyelvóra, ha a gyakorlást a számítógép segíti?
IGEN = 16 fő NEM = 1 fő

Miért? Mert segít, mert az óra nem idegen légkörű, mert él az osztály.

TÓTH ETELKA

A programot az OKTA GMK készítette és forgalmazza. Kapható még a KÖNYVÉRTÉKA Áruházban (Bp. V., Báthori u. 9.) és az Október 6. utcai tankönyvboltban. A programot kazettán árulják (ára 500,— Ft), kívánságra az OKTA GMK floppyra is átmásolja. Az OKTA GMK címe: 1476 Bp. Pf. 44.

HELYREIGAZÍTÁS: Sajnos a 9. számban a **Fonalfék, fonalerő** című cikk három képleténél a középre helyezésre utaló jeleket is kiszedték, amelyek így természetesen értelemzavaróak. E cikk 4. és 5. ábrája fel van cserélve. A hibákért olvasóink elnézését kérjük.

Nemzetközi számítástechnikai programozási verseny Bulgáriában

Első alkalommal rendeztek nemzetközi számítástechnikai versenyt 18 évnél ifjabbak számára. A gondolat magyar fejekből pattant ki, de a házigazda szerepére Bulgária vállalkozott. Ebben a szerepben a rendezők igazán jól vizsgáztak: mindent megtettek a „viadal" jó szellemű és igazságos lebonyolítása érdekében.

A feltételek és körülmények

A magyar csapatot az NJSZT Ifjúsági Bizottsága állította össze. A részvétel kivívásáért előzetes erőpróbán kellett részt venniük. A válogató alapján Farkas Károly, Schadt György, Kovács László és Boros Péter képviselte a magyar színeket Szófiában. Őket Zsakó László és Csopaki Gyula kísérte el, akik egyben a nemzetközi versenybizottság tagjai is voltak.

A versenyt eredetileg két kategóriában írták ki: 18 és 16 év alattiak számára. Hogy végül lett egy harmadik kategória is, annak egy igazságos döntés az oka: a román csapat két 14 év alatti versenyzőt is hozott magával. Őket természetesen külön értékelték. Egyébként Bulgária kettő, Csehszlovákia, az NSZK, Magyarország, Románia és a Szovjetunió egy-egy csapattal indult.

A háromnapos rendezvény nyitányaként a résztvevők megtekintették a verseny színhelyét, és kipróbálták a gépeket. A rendezők, amint ezt előre közölték, IBM PC klónokat és Apple II gépeket készítettek elő. A román csapat ragaszkodott a ZX-Spectrumhoz. Ennek sem volt semmi akadálya.

A verseny

A megérkezést követő napon a gyerekek tovább ismerkedtek a gépekkel, amíg a felnőttek a feladat kiválasztásán törték a fejüket. Minden ország hozott egy feladatot. Hosszas töprengés után a bolgár szervezők által összeállított feladatot fogadták el. Ez, amint azt olvasóink is láthatják, kifejezetten matematikai feladat számítógépen való megoldása. Zsakó László elmondása szerint a csehszlovák és a magyar csapat számítástechnikai alkalmazási feladatokat terjesztett elő, amelyeket a rendezők könnyűnek találtak és eleve kizártak. Szó sem lehet részrehajlásról, hanem úgy tűnik, hogy a bolgár szakemberek a számítástechnika és a matematika bizonyos részeinek összefonódását tekintik a legkiforrottabbnak.

A feladat kiválasztásától kezdve a résztvevők nem találkozhattak a versenybizottság tagjaival. A 14-től 18 óráig tartó „futam" után a kiértékelés talán még nagyobb feladatot jelentett, mert a zsűri hajnali 2 órakor fejezte be munkáját. Eközben a gyerekek ismerkedtek egymással, beszélgettek. Nyelvi nehézségük nem volt, angolul, németül és „számítástechnikául" megértették egymást.

A befejező napon a nyertesek — az első helyezettek a hazai pálya előnyét élvező bolgár versenyzők lettek — a Kongresszusi Központban ünnepélyes keretek között kapták meg a megérdemelt elismerést és az oklevelet.

A versenyen a fiatalok bizonyították felkészültségüket és tudásukat. Nem lebecsülendő, hogy nem okozott problémát az MS-DOS használata, pedig középiskoláinkban ez nem igazán elterjedt. A vetélkedők közül volt, aki C nyelven, más pedig Pascal nyelven írta az erőmérő programot.

Tanulságok

A befejezés után szokás a mérleget megvonni. Ez most sem hiányozhat, bár a feladat, amit közlünk, nem ad lehetőséget a részmegoldások értékelésére. Aki tisztában volt a matematikai megoldással — amit ismereteim szerint ebben a korban nem tanítanak —, előnyben volt. Akinek pedig a verseny során kellett kitalálnia, nem maradt ideje a program megírására. Továbbá az elbírálás lehetőségére is tekintettel kell lenni. Nem jó, ha a versenybizottságot szinte megoldatlan feladat elé állítják. (Reméljük, hogy a következő nemzetközi ifjúsági számítástechnikai programozási versenyt Magyarország rendezi meg!)

A magyar versenyzők, bár nem nyertek, mégsem szomorúan tértek haza. A matematikai részt megoldották, kitalálták, és még részprogramra is jutott idejük. Kitűnni nem sikerült, de helytálltak.

A feladat

Az N halmaz tartalmazza egy város buszmegállóinak sorszámait:
$N = \{1,2,3, \ldots n\}$ $n \leq 20$
Az M_j sorozatok tartalmazzák az egyes buszok megállóinak sorszámait:
$M_1 = (i_{1\,1}, i_{1\,2}, \ldots i_{1\,m_1})$
$M_2 = (i_{2\,1}, i_{2\,2}, \ldots i_{2\,m_2})$
.
.
.
Mj
.
.
.
$M_r = (i_{r1}, i_{r2}, \ldots i_{r\,m_r})$
ahol $r \leq 10$ és i_{jk} az N halmaz eleme, továbbá ha $k \neq 1$, akkor $i_{jk} \neq i_{j1}$. Minden M_j a buszmegállók egyik menetirányban megadott felsorolása, de természetesen a buszok mindkét menetirányban ugyanazokban a megállókban állnak meg. Tervezz programot a következő specifikációk alapján: Olvasd be az n és r egész számokat és az $M_1, M_2, \ldots M_r$ vagyis az M_j sorozatok elemeit!

Részfeladatok

A) Ellenőrizd, hogy valaki el tud-e jutni bármelyik buszmegállóból bármelyik buszmegállóba, a választ írd ki a képernyőre (igen, nem)!

B) Olvasd be két megálló sorszámát ($p,q \leq 20$), keresd meg a két megálló közötti ciklus (zárt hurok) nélküli összes utat, írd ki az utakat az érintett buszmegállók felsorolásával!

C) Olvasd be két megálló sorszámát ($s, t \leq 20$), keresd meg e megállók között a leggyorsabb útvonalat! Egyenlőnek tekintjük bármely két szomszédos megálló közötti utazási időt, és minden átszállás háromszor annyi időt vesz igénybe, mint a két megálló közötti utazási idő.

Megjegyezzük, hogy a versenyen közzétették: a 16 éven aluli versenyzőknek az A feladatot nem kellett megoldaniuk, e feladat megoldása csak holtverseny esetén számított volna az értékelésnél.

PINKE GYÖRGY

Magyar nyelvű BASIC C64-re

A program segítséget nyújt azoknak is, akik a BASIC nyelvet nem ismerik, de szeretnének számítógépükkel kommunikálni. És alkalmas lehet még másra, mint például bemutatásra is. Mivel a magyar nyelvű interpreter a BASIC használta memóriaterületet nem veszi igénybe, lehetséges bármi-

Utasítás	Magyar megfelelő	Rövidítés	Utasítás	Magyar megfelelő	Röv.
end	vege	vE	new	uj	uj
for	szamolj	sZ	sgn	sgn	sG
next	kov	kO	int	int	iN
data	adat	aD	abs	abs	aB
input	be	be	usr	usr	uS
input #	be #	bE	fre	hely	hE
dim	tomb	tO	pos	poz	pO
read	olvass	oL	sqr	gyok	gY
let	!	!	rnd	vel	vel
goto	menj	mE	log	log	lO
run	fuss	fU	exp	exp	eX
if	ha	ha	cos	cos	cO
restore	ujra	uJ	sin	sin	sI
gosub	szubr	szU	tan	tan	tan
return	vissza	vI	atn	atn	aT
rem	megj	meG	or	vagy	vaG
stop	allj	aL	and	es	es
on	fugg	fuG	step	lepes	lE
wait	varj	vA	not	nem	nE
load	tolts	toL	then	akkor	aK
save	ments	menT	spc(	ur(	uR
verify	ell	eL	fn	fn	fn
def	def	dE	to	tol	tol
poke	tar	tA	tab(	tab(	taB
print	irj	?	mid$	kozep$	koZ
print #	irj #	iR	right$	jobb$	joB
cont	folyt	fO	left$	bal$	bA
list	lista	lI	chr$	kar$	kA
clr	tor	tor	asc	kod	kod
cmd	cel	cE	val	ert	eR
sys	gepi	gE	str$	str$	sT
open	nyiss	nY	len	tag	tag
close	zarj	zA	peek	les	les
get	fogd	foG			

```
0 REM ATOMSOFT 1987
5 PRINT"■■■          TEKEM VARJON"
10 FOR I=49408 TO 49151
20 POKE I,PEEK(I):NEXT
30 FOR I=57344 TO 65535
40 POKE I,PEEK(I):NEXT
50 POKE 1,53
60 POKE 56292,141
70 POKE 56294,142
80 POKE 56293,14
90 POKE 56295,4
100 POKE 56540,65
110 POKE 56541,143
120 POKE 56549,29
130 POKE 56550,14
140 POKE 41848,17
150 POKE 41849,75
160 POKE 41850,69
170 POKE 41851,33
180 POKE 41852,96
200 A$="VEG SZAMOL KUSZABAI BE B TOMIOLVAS■$MEN FUS■HAUJR■SZUB VISSZ■MEG ALL "
210 A$=A$+"FUGGVENY TOLT■MLNT■ELLDE TA IRJ IR FOLYILIST■TO CELGEP NYIS■ZAR FOG"
220 A$=A$+"U TAB DLF UR■KKOLNE LEPE■++ L"+CHR$(222)+"E■VAG ↑ ←"
230 A$=A$+"SO INTRO■US HEL IPO■UYO VELLOI EX CO■SI TA AT LE■TAI STR ERI "
240 A$=A$+"KO KAR BHL JODE KOZEP "
250 FOR I=1 TO 254
260 POKE 41112+I,ASC(MID$(A$,I,1))
270 NEXT
300 A$="TUL NAGY FILE■NYITOTT FIL ZART FIL TA FILE NEM LETEZIK■"
310 A$=A$+"AZ ESZKOZ NEM LETEZIK■NEM BE-FIL NEM KI-FIL NIANYZO FILENEV■"
320 A$=A$+"NEM JO EGYSEGSZAM■SZAMOLJ NELKULI KOMSZINTAKTIKA SZUBP NELKULI "
330 A$=A$+"VISSZ■MADIK TOLVARAS ILLEGALIS SZA TULCSORDULAS MEMORIA■NEM LETEZO SOR■"
340 B$="TOMB NAGYSAG UJRADIMENZIONALAS■OSZTAS 0-VAL■NEM DIREKT UTASITAS■"
350 B$=B$+"STRING-SZAM HOSSZU STRING FILE ADAT TUL BONYOLULT■"
360 B$=B$+"NEM FOLYTATHATO■DEF INTEGER ELL.-TOLTES"
370 FOR I=1 TO 244
380 POKE I+41377,ASC(MID$(A$,I,1))
390 NEXT
400 FOR I=1 TO 150
410 POKE I+41617,ASC(MID$(B$,I,1))
420 NEXT
430 POKE 813,165
440 DATA 158,161,172,161,184,161,193,161,212,161,234,161,245,161,0,162,16,162
460 DATA 24,162,53,162,65,162,85,162,97,162,111,162,124,162,131,162,146,162
470 DATA 158,162,176,162,189,162,209,162,229,162,231,162,242,162
480 DATA 0,163,16,163,21,163,42,163
490 FOR I=1 TO 56
500 READ A:POKE 41767+I,A:NEXT
510 POKE 41829,74:POKE 41830,29
520 POKE 41835,74:POKE 41836,73
530 POKE 41837,66:POKE 41838,65
540 POKE 41839,32:POKE 41842,58
550 POKE 41843,32:POKE 41859,32
560 POKE 41860,65:POKE 41861,76
570 POKE 41862,76:POKE 41863,74
580 A$="SOK AZ ADAT"+CHR$(13)+CHR$(0)+" -HIBAS INFORMACIO"+CHR$(13)+CHR$(0)
590 FOR I=1 TO 34
600 POKE I+44283,ASC(MID$(A$,I,1))
610 NEXT
620 A$=CHR$(13)+"KERESEM"+CHR$(160)+"    "+CHR$(160)
630 A$=A$+CHR$(13)+"-NYOMJON PLAY-T !"+CHR$(160)+"    -NYOMJON PLAY-T ES REC-OT"
640 A$=A$+CHR$(160)+CHR$(13)+"TOLTES"+CHR$(160)+CHR$(13)+"MENTES"+CHR$(160)
650 A$=A$+CHR$(13)+"KONTROL "+CHR$(160)+CHR$(13)+"VAN "+CHR$(160)+CHR$(13)+"JO"
660 FOR I=1 TO 97
670 POKE I+61640,ASC(MID$(A$,I,1))
680 NEXT
690 POKE 60647,84:POKE 60649,204
700 POKE 60650,32:POKE 60652,70
710 POKE 60653,213:POKE 60654,32
1000 PRINT"■■          MAGYAR NYELVU BASIC"
1010 PRINT"■■    COMMODORE 64  *  ATOMSOFT 1987"
1020 PRINT"■■        38911 BASIC BYTES FREE"
1030 NEW
```

lyen, már meglevő program futtatása. Így lefordítható CBM BASIC-röl, hiszen valamennyi utasítás, hibaüzenet stb. magyar nyelvű. (Az utasítások megfelelői és rövidítései a táblázatban megtalálhatóak.) A program használatakor ezek az utasítások magyarul jelennek meg, s teljesen kompatibilis a CBM BASIC-kel. A billentyűzet használata nem változott meg, de megváltoztatható. Például a Z és Y cseréje a következőképpen történhet: TAR 60314,90: TAR 60301,89/. Az f gombok jelentése a következő: f1-képernyőtörlés, f3-nagybetűk/grafikus jelek, f5-kisbetűk/nagybetűk, f7-SHIFT RETURN.

A program begépelésekor ügyeljünk a pontos beírásra. Indítás után mintegy másfél percet várni kell, amíg a szükséges változások elkészülnek. Ha az eredeti CBM BASIC-et akarjuk használni, akkor csak a STOP—RESTORE billentyűket kell megnyomni. Magyar nyelvre újfent visszatérni a POKE 1,53 parancs kiadásával tudunk.

BÁRTFAI BARNABÁS

Spectrumra írt bináris átalakító

A ZX—Spectrum BASIC utasításai között megtalálható a BIN függvény, s ez az argumentumként megadott bináris számot decimális formába alakítja át. Sajnos hiányzik az inverze. Mivel a programozás közben többször szükség lehet arra, hogy egy-egy karakter bittérképét szemügyre vegyük, az 1. lista kis rutinja ezért készült.

A program begépelése és futtatása után a 0 és 255 közé eső számot egyszerű POKE utasítással helyezzük el a memóriába, a 65500-as címre (16 k-s gépen 32001-re), majd RANDOMIZE USR 65000, (32000 a 16 k-s gépen) utasítással futtassuk.

Azoknak, akiknek nincs assemler fordítóprogramjuk, ajánljuk a BASIC betöltőt, a 2. számú listát.

Tetszőleges karakter bittérképének kiírására mutat példát a 3. számú lista.

CSAJTAI KORNÉL

```
BIT     LD C,n          ; C-be a szám
        LD B,8          ; ciklus számláló
C1      BIT 7,C         ; legnagyobb helyi-
        LD A,48         ; értékű bit
        JR Z,KI0        ; ha 0, akkor 0-t ír
        INC A           ; ha 1, akkor 1-et
                        ; ír ki
KI0     RST 16
        RLC C           ; C-t elforgatja
        DJNZ C1         ; vissza a vizsgálat-
                        ; hoz
BREAK   LD A,(23560)    ; mivel a rutin a
        CP 32           ; kommunikációs kép-
        RET Z           ; ernyőre ír, így a
                        ; BREAK-re vár a
        JR BREAK        ; visszatéréshez,
                        ; elkerüli a szám
                        ; törlődését.
```

1. lista

```
10 CLEAR64999:REM 16 K-n 31999
20 FORi=0TO23
30 READa:POKE65000+i,a:REM 16K-n 32000+i
40 NEXTi
50 DATA 14,  0,  6,  8,203,121, 62, 48
60 DATA 40,  1, 60,215,203,  1, 16,244
70 DATA 58,  8, 92,254, 32,200, 24,248
```

2. lista

3. lista

```
10 INPUTa$:LETa=CODEa$:FORi=0TO7:POKE
   65001,PEEK(15360+8*a+i):RANDOMIZE
   USR65000:PAUSE0:CLS:NEXTi
20 GOTO10
```

Spectrum reflex!

A játékos feladata, hogy a ‚FIGYELJ' feliratot követő négyzet megjelenése után, minél hamarabb megnyomjon egy gombot. A gép – belső óráját felhasználva – értékeli a reakcióidőt. A 60-as sor utasításai a belső óra nullázására szolgálnak. Íme a program listája, mely azonban nem Spectrum gépen készült.

SZARKA DÁNIEL

```
  1 LET H$="Spectrum": LET hi=5
 10 LET a=INT (RND*100): LET csal=0
 12 PRINT AT 3,11;FLASH 1;"FIGYELJ!"
 15 PAUSE 100
 20 IF a=0 THEN GOTO 10
 30 PAUSE a
 40 IFINKEY$ <>"" AND csal <1 THEN GOTO
    1000
 50 PRINT AT 10,13; "     ";
 51 PRINT AT 11,13; "     ";
 52 PRINT AT 12,13; "     ";
 53 PRINT AT 13,13; "     ";
 54 PRINT AT 14,13; "     "
 55 REM az 50-54 sorok idézőjelei közt
 56 REM 5 db grafikus 8-as szerepel!
 60 POKE 23672,0: POKE 23673,0
 61 POKE 23674,0: REM belső óra 0-ra
 70 IF INKEY$<>"" THEN LET ido=(PEEK
    23672+256*PEEK 23673+65536*PEEK
    23674)/50: GOTO 85
 80 GOTO 70
 95 CLS
100 PRINT AT 1,1;"JATEKOS";TAB10;
    "IDEJE"
110 PRINT: PRINT
130 PRINT AT 3,11;ido
131 PRINT AT 20,5;"CSUCS:";hi;
132 PRINT " Tartja:";H$
135 PAUSE 0
140 IF ido<hi THEN LET hi=ido: GOTO 200
150 INPUT "JATSZUNK MEG?";K$
160 IF K$="i" OR K$="I"THEN CLS: GOTO 10
170 CLS: PRINT AT 10,8;FLASH 1;"SZIA":
    STOP
200 CLS: PRINT AT 10,11;"EZ REKORD";
    AT 11,4;"IRD BE A NEVED! (1-4)":
    INPUT H$
205 IF LEN H$>4 THEN GOTO 200
210 LET ido=0: CLS: GOTO 150
1000 CLS: PRINT AT 10,12;"CSALTAL!":
    GOTO 150
9000 SAVE "REFLEX" LINE 9500: VERIFY"":
    STOP
9500 CLS: PRINT AT 10,10;"STOP THE
    TAPE!": PAUSE 100: CLS: GOTO 1
```

DIÁK C64 ROVAT

Mastermind C64-re

(Lapunk 5. számában írtunk már egy mastermind programot, az alább közölt munka ennek variánsa. A szerk.)

A játék és a program rövid ismertetése

A lényeg, hogy az adott hosszúságú — a beküldött programban a 0-9 számjegyeket akár többszörösen is tartalmazó — karakterláncot, tippek alapján, minél kevesebb lépésből találja ki a játékos. Következő tippjének kialakításában az ellenfél — a gép — segíti. Mindig megadja az előző próbálkozás eredményét. Fehér — a programban WH — értéke az alaki és helyiértékre is eltalált számjegyek számát adja, míg a fekete — BL — a csak alaki értékre eltalált számjegyek darabszámát mutatja meg. A játék során lehetséges találgatások száma is korlátozott!

A program egyes részeinek, fontosabb, módosítható változóinak magyarázata:

10—12 sorok az alap beállításokat végzik,
20—23 sorok az aktuális játszmát inicializálják,
30—34 sorok a tipp bekérését és a konverziót bonyolítják,
40—43 sorok a sikertelen játszmát kezelik,
50—53 sorok a játék folytatását garantálják,
60—65 sorok új játék kérését adják,
80—83 sorok a kurzorpozicionáló rutinok,
90—94 sorok karakterlánc konverziót hajtanak végre,
100—130 sorok adják a játék lényegét.

A 10-es sor HOSSZ változója a kitalálandó szám hosszát tartalmazza, a képernyőkép elrontása nélkül, 1—28 értékeket vehet fel. Az egy játékon belüli tippelések maximális számát a 11-es sor MA változója adja meg, értéke szabadon választható, adott „szóhossznál" a nehézségi fokozatkapcsoló szerepét játszhatja, illetve „reménytelen" esetekben vészkijáratot ad. A győztes-vesztes játékok kiírásában megjelenő, „bőbeszédű" szövegek lényeges szerepe, hogy a maximális szóhossznál is töröljék az utolsó beadott tipp értékét. Újabb tipp beírásánál a kurzorvezérlő, törlő, beszúró billentyűk szabadon használhatók, sőt RETURN válaszra az előző tipp marad érvényes, csak a próbálkozások száma nő.

FÖLDI ENDRE

```
10 PRINTCHR$(147):HOSSZ=04:REM HO=01-28
11 DIMSA(HO),GE(HO):A$="000":MA=99
12 FORC=0TO38:SP$=SP$+" ":NEXT
20 NR=0:SA$=""
21 FORC=1TOHO:GE(C)=INT(RND(1)*10):NEXT
22 GOSUB80:GOSUB81
23 PRINT" NR WH BL  TIPP"
30 GOSUB83:PRINTTAB(10)SA$
31 GOSUB83:INPUT" A TIPPED";A$
32 SA$=" ":FORC=1TOHO
33 A=VAL(MID$(A$,C,1)):SA(C)=A
34 SA$=SA$+CHR$(48+A):NEXT
40 NR=NR+1:IFNR<=MATHEN50
41 GOSUB81:GOSUB80
42 PRINT" MOST AZ EGYSZER NEM TALALTAD";
43 PRINT" EL!!!!!!!":GOTO62
50 GOSUB100:H=3:A=NR:GOSUB90:NR$=A$
51 H=2:A=WH:GOSUB90:WH$=A$:A=BL:GOSUB90
52 GOSUB80:PRINTNR$+" "WH$+" "A$+"  "SA$
53 IFWH<>HOTHEN30
60 GOSUB81:PRINT" GRATULALOK!!!";
61 PRINT" NYERTEL!!! ELTALALTAD!!!"
62 GOSUB83:INPUT" JATSZOL MEG";A$
63 IFASC(A$)=73THEN20:REM 73=I ASC KOD
64 END
65 :
80 GOSUB83:PRINTSP$:REM INPUT MEZO TORL
81 POKE214,24:SYS58732:PRINT:REM SCROLL
82 POKE214,21:SYS58732:RETURN
83 POKE214,23:POKE211,0:SYS58732:RETURN
90 A$=MID$(STR$(A),2,H)
91 A$=" "+RIGHT$(A0$+A$,H):RETURN
92 :
93 :
100 REM **********************
101 REM * A PROGRAM "LELKE" *
102 REM **********************
105 WHITE=0:BLACK=0
110 REM FEHER KISZAMITASA
111 FOR GE=1 TO HO
112 IFGE(GE)=SA(GE)THENWH=WH+1:SA(GE)=-1
113 NEXTGE
120 REM FEKETE KISZAMITASA
121 FORGE=1TOHO
122 IFSA(GE)=-1THEN126
123 FORSA=1TOHO
124 IFGE(GE)=SA(SA)THENBL=BL+1:SA(SA)=-2
125 NEXTSA
126 NEXTGE
130 RETURN
```

Pályázati felhívások

Az 1988 márciusában megrendezendő III. Nemzetközi Mikroszámítógépes Találkozó keretében meghirdetjük

az oktatóprogramok

egyéni és csapat országos versenyét. A kiírt kategóriák:

Biológia — Kémia
Fizika
Humán (két különböző témájú)
Matematika
Nyelv
Szakirányú (két különböző témájú)
Technika
Egyéb

Az egyéni versenyben — kisvállalkozások és vállalatok kivételével — bárki részt vehet. Egy kategóriában minden pályázó csak egy programmal szerepelhet.

A csapatversenyben iskolák indulhatnak általános iskola, szakmunkásképző, szakközépiskola, gimnázium és felsőoktatás

DIÁK HOMELAB ROVAT

Több szólam Homelabra

A ZX—Spectrum hangjait ültettük át szoftverből Homelabra. A három kis rutinnal két-, három és négyszólamú zene készíthető. Itt csak a két szólamot szimuláló zene programját közöljük, s nem többet. Ugyanis a két másik rutin ugyanazon az elven működik.

A programok assembly nyelven íródtak, nagy előnyük, hogy a memóriában bárhová helyezhetők. Beírásuk monitorban egyszerű. Természetesen akiknek van assemblere, az abban is beírhatja. Mindhárom rutin futásakor a háttér-regisztereket és az IX regisztert használja. A két szólamot szimuláló program (1. lista) részletesen:

- a hangok időtartama a D regiszterbe kerül,
- az első hang magassága H,
- a másodiké D,
- figyelem! Minél magasabb hangot akarunk, annál kisebb értéket kell a kiválasztott regiszterbe tenni. Ez persze nem vonatkozik a tartamra, noha az függ a hangok magasságától,
- a program a hangadásnál tiltja a képernyőt, ezért nem látszik a kép.

A 3 szólamot szimuláló program a 2. listán, a 4 szólamot szimuláló a 3. listán található.

A legszebb hangzás a két szólamot utánzó rutinnal érhető el, hiszen a másik kettő már jobban megközelíti a gép teljesítőképességének határát.

Jelmagyarázat a listákhoz:

- T tartam
- L első szólam hangmagassága
- M második hangmagasság
- H harmadik hangmagasság
- N negyedik hangmagasság

Az 1. listán a számok előtt álló H=$ jel a hexa értéket jelenti.

ARANYI FERENC

1. lista

```
KEZD 32003E MV(H3E00),A;KéPERNYőT TILTJA
     D9     EXX
     16T    LD D,T;D-BE AZ IDőTARTAM
VISZ DD29   ADD IX,IX
     25     DEC H
     2009   JRNZ ;IDE
     0603   LD B,H03
FO 1 32003C MV(H3C00),A;1.HANG MEGSZOLAL
     10FB   DJNZ FO 1
     26L    LD H,L ;1.HANG MAGASSAG
IDE  DD29   ADD IX,IX
     2D     DEC L
     20ED   JRNZ VISZ
     0603   LD B,H03
FO 2 32003C MV(H3C00),A;2.HANG SZOL
     10FB   DJNZ FO 2
     2EM    LD L,M ;2.HANG MAGASSAG
     15     DEC D
     20E1   JRNZ  VISZ
     D9     EXX
     32003F MV(H3F00),A;KéP VISSZA
     C9     RET
```

2. lista

```
KEZD 32 00 3E D9 16 T DD 29
     25 20 09 06 03 32 00 3C
     10 FB 26 L  DD 29 2D 20
     09 06 03 32 00 3C 10 FB
     2E M  DD 29 0D 20 DF 06
     03 32 00 3C 10 FB 0E H
     15 20 D3 D9 32 00 3F C9
```

3. lista

```
KEZD 32 00 3E D9 16 T  DD 29
     25 20 09 06 03 32 00 3C
     10 FB 26 L  DD 29 2D 20
     09 06 03 32 00 3C 10 FB
     2E M  DD 29 0D 20 09 06
     03 32 00 3C 10 FB 0E H
     DD 29 1D 20 D1 06 03 32
     00 3C 10 FB 1E N  15 20
     C5 D9 32 00 3F C9
```

---

csoportban. Minden csapatnak legalább öt különböző kategóriában kell előre benevezni. Egy iskola több csapattal is versenyezhet. A csapat tanárokból és diákokból állhat, de legalább egy tanulónak önálló programmal kell jelentkeznie.

Jelentkezési határidő: 1988. február 22.

a házi építésű számítástechnikai eszközök

országos versenyét

rendszerek,
perifériák és
számítógéppel vezérelhető modellek
(például modellvasút, autó robot stb.)

kategóriában.

Jelentkezési határidő: 1988. február 22.

a sütés — főzés számítógéppel

című országos vetélkedőt.

A versenyen azok indulhatnak, akik valamilyen módon felhasználják a számítógépet ételek készítéséhez (például recept-összeállítás, vezérlés).

Jelentkezési határidő: 1988. február 22.

Jelentkezési lap, valamint a pályázatokról részletesebb információ az NJSZT titkárságán (Bp. V., Báthori u. 16.) szerezhető be.

A versenybizottság

OKTA-TOTÓ

A strukturált programozás az egységenként kezelhető programrészek használatán alapuló technika. Lehetővé teszi a program összeállításának elosztását több program között, illetve egyes szerkezetek különböző programokban való alkalmazását. **(2)**

A multiprogramozás több program egyidejű indítását teszi lehetővé. Ezzel a számítógép részegységeinek optimális kihasználását lehet biztosítani. Az egyidejű végrehajtás látszólagos, mert a központi egységben csak egy program fut. **(X)**

A kiegészítő aritmetikai processzor meggyorsítja a matematikai műveletek végrehajtását. Különösen a trigonometrikus függvények esetében célszerű használni. Eléréséhez speciális fordítóprogram szükséges **(1)**, így a negyedik kérdés megoldása: **2.**

Ha a programban sokszor ismétlődő rész van, azt célszerű szubrutinként felírni. Ezzel jelentős mértékben csökkenthető a program mérete. Vigyázni kell azonban, hogy véletlenül ne hajtódjon végre. **(X)**

Olyan tár, amelyből a legutoljára bevitt adat érhető el először. Kialakítása szerint lehet hardververem és szoftververem. **(1)**

A ciklusok egymásba ágyazása, amikor cikluson belül újabb ciklust szervezünk. Mindig a belső „beágyazott" ciklus hajtódik először végre. **(X)**

A ciklusok maximális egymásba ágyazhatósága a fordítóprogramtól függ. Mivel a fordítóprogram általában géptípushoz kötődik, ezért a géptől is. Minden esetben meg kell nézni a fordítóprogram és a gép kézikönyvét. **(1)**

Logikai elágaztatással szervezhető ciklus, például amikor valamely változó értékének függvényében, illetve a feltétel teljesülése esetén a vezérlést egy előző utasításra adjuk át. Szervezése nehézkes és megnöveli a program méretét. **(2)**

Összegzésnél meg kell adni a változó kezdőértékét, illetve egyenlővé kell tenni nullával. Igaz, egyes géptípusoknál minden változó kezdőértéke nulla, de a biztonság kedvéért erre ne alapozzunk. **(X)**

A tömbök maximális dimenziója géptől és programnyelvtől függ. Kétdimenziós tömböt általában minden gépen lehet definiálni. **(2)**

A 12. kérdésre a válasz: igen. Mátrixműveletek esetében ez szükséges is. **(1)**

Az aritmetikai műveletek pontosságát növelni lehet a változó definiálásával. Például duplapontosságúnak definiáljuk. **(2)**

Az EXE kiterjesztésű fájl nevének begépelése után közvetlenül futtatható. Egyes fordítóprogramok automatikusan képezik. **(1)**

1	2	3	4	5	6	7	8	9	10	11	12	13+1
2	X	1	2	X	1	X	1	2	X	2	1	2	1

Rejtvényjátékunk az utolsó fordulójához érkezett. Akik figyelemmel kísérték, tudják, hogy a pályázati szelvény helyes kitöltői között öt darab 200 forint értékű könyvutalványt sorsolunk ki.

Azok az olvasók, akik mind a hat szelvényt jól töltötték ki, egy kis szerencsével számítógépet nyerhetnek.

Az utolsó OKTA-TOTÓ beküldési határideje: **1987. október 18.** A három megadott válasz közül csak az egyik helyes. A tippeket (1, 2 vagy X) a pályázati szelvényre kell rávezetni. Szelvény nélkül érkezett megoldásokat nem fogadunk el. Kérjük, hogy a pályázati szelvényt postai levelezőlapra ragasztva küldjék el a szerkesztőség címére: 1317 Budapest, Pf.: 443.

Az első OKTA-TOTÓ sorsolásán az alábbi olvasóinknak kedvezett a szerencse:

Sebők Mihály	Nyíregyháza, Gerhát u. 8. 4400
Gáthy Attila	Berettyóújfalu, Bessenyei-ltp. 2. 4100
Naszódi László	Budapest, Ferenchegyi út 29. 1025
Pásztor László	Bonyhád, Kodály Zoltán u. 13. 7150
Buda István	Budapest, Hajdú u. 19. 1139

A hatodik forduló kérdései:

1. A számítógép operációs rendszer nélkül
1. nem üzemeltethető
2. a gép öntesztelése meg sem indul
X. a gép működik, de utasításokra nem reagál

2. Merevlemez nélül (Winchester) lehet-e operációs rendszert alkalmazni?
1. nem, mert nem lehet betölteni
2. hajlékonylemezről betölthető
X. beolvasható, de nem lehet a parancsot végrehajtani

3. Mi a Boole-algebra?
1. A Bolyai által feltalált algebra továbbfejlesztett változata
2. a "boot", behúzás szóból ered, és adatátvitelt jelent
X. az igaz és hamis állításokon felépített kételemű algebra

4. Melyik nem felel meg hexadecimális számnak a számítástechnikai jelölésrendszerben?
1. 17
2. F63
X. 56G

5. Melyik a helyes növekvő sorrend az adatbázis kialakítása szempontjából?
1. rekord, fájl, mező
2. mező, rekord, fájl
X. fájl, rekord, mező

6. Mire szolgál az ellenőrzőszám?
1. az esetlegesen hibásan bevitt adatok kiszűrésére
2. illetéktelen személy ne tudja ismerete nélkül elindítani a számítógépet
X. az aktuális adatbázis lehívására

7. Mi a Chapin kártya?
1. a grafikus megjelenítést vezérlő chip
2. a folyamatábra helyett alkalmazható programozási eszköz
X. speciális anyagból készült, nagy integráltságú interfész

8. Mi a jelentősége a bináris számábrázolásnak a számítógép felépítésénél?
1. semmi, a 16 és 32 bites mikroprocesszorok ezt túlhaladták, nem használják
2. a félvezetők igen (1) és nem (0) jelet generálnak, ez az alapja minden műveletnek
X. az IC kiváltotta az elektroncsövet, ezzel megszünt a felhasználása

9. Indexelt szekvenciális adatállomány melyik adathordozón található?
1. mágneslemezen
2. mágnesszalagon
X. minden szekvenciális fájlt fogadó periférián

10. Mit csinál a disassembler?
1. az assembler memória területet formázza
2. helyesen assembly, perifériákat vezérel
X. gépi kódból assembler nyelvű programot állít elő

11. Mi az algoritmus?
1. olyan szabályok, amelyek egy feladat végrehajtásának egymást követő lépéseit határozzák meg
2. számítástechnikai folyamat, amely egy feladat programozását teszi lehetővé
X. a kibernetikában a leírható folyamatokra alkalmazott kifejezés

12. Mi a PS/2?
1. a mikroszámítógépek szabványosított neve
2. az új IBM gépcsalád neve
X. a második generációs személyi számítógépek

13. Milyen különbséget tesz a számítógép a parancs és az utasítás között?
1. a parancsot bevitelének pillanatában végrehajtja, amíg az utasítást a program futása közben
2. semmilyet, rokonértelmű szavak
X. parancsnál nem kell argumentumot megadni, de az utasítást annak függvényében hajtja végre

+1. Milyen célt szolgál a "DOS" parancs?
1. vissza lehet térni vele az operációs rendszerbe a program befejezése nélkül
2. a merev- és hajlékonylemez-meghajtót vezérli
X. áttér a rendszer a BASIC interpreterből más fordítóprogramba

PÁLYÁZATI SZELVÉNY 87/10

OKTA-TOTO

Kérdés	Tipp
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
+	
13+1.	

Név: _____

Cím: _____

Sz. szám:

BASIC és gépi kód

Legutóbb a logikai utasításokról volt szó. Most két újabb utasítással ismerkedünk meg.

A BIT utasítás

Neve az angol BIt Test kifejezésből származik. Az A regiszter és az operandus által meghatározott memóriabájt tartalmával logikai ÉS műveletet végez, és az eredménynek megfelelően állítja be a Z, V és N feltételbiteket. Az AND utasítástól abban különbözik, hogy az A regiszter tartalma a BIT végrehajtása során nem változik meg. Ezenkívül a V feltételbitet is beállítja, az eredmény 6. bitjével megegyező értékre.

A BIT utasítás csak kétféle címzési móddal használható: abszolúttal és nullalaposssal. A feltételbitek beállításán kívül egy különleges alkalmazása is van, ezt a programlisták segítségével mutatom be.

A három listán a C64 ROM-jának ugyanaz a részlete látható, csak a kezdőcímben különböznek. Az itt látható rutinokról „Az ismeretlen C16" című sorozatban részletesen írtam (Az ismeretlen C16 — Rendszerváltozók, beépített rutinok; Mikroszámítógép Magazin 1986/7.), a CHKCOM rutinról ($AEFD) pedig már jelen sorozatban is többször volt szó. A $AEFF címen kezdődő rész vizsgálja, hogy a BASIC szöveg aktuális karaktere megegyezik-e az A regiszter tartalmával. Ha nem, akkor a $AF08 címén kezdődő hibarutinnal folytatódik a futás.

Ha a csukó zárójel jelenlétét akarjuk vizsgálni, a JSR $AEF7 utasítással hívjuk meg az ellenőrző rutint. Ekkor a csukó zárójel ASCII kódja az A regiszterbe kerül, majd két BIT utasítás hajtódik végre. Beállítják a feltételbiteket, de ezt nem használjuk ki, a lényeg az, hogy minden más változatlan marad. JSR $AEFA utasítással hasonlóképpen vizsgálhatjuk, hogy az aktuális karakter nyitó zárójel-e. Itt csak egy BIT utasítást kell elvégezni a fenti módon, a CHKCOM hívásakor pedig már egyet sem.

Ilyen módon egyszerűen oldható meg, hogy valamely rutinnak több belépési (idegen szóval entry) pontja legyen, melyek csak kismértékben módosítják a rutin működését. Jelen esetben eltérés csak az A regiszter tartalmában van. Hasonlóan használhatjuk az utasítás nullalapos változatát is.

A NOP utasítás

Neve az angol No OPeration kifejezésből származik, jelentése: nincs művelet.

Üres utasításnak is nevezik, ugyanis végrehajtásakor az utasításszámláló regiszter tartalmán kívül semmi sem változik. Ennek ellenére többféle célra használják, alkalmazásával gyakorlati példák keretében ismerkedünk majd meg.

BARNA LÁSZLÓ

```
aef7  a929      lda #$29
aef9  2ca928    bit $28a9
aefc  2ca92c    bit $2ca9
aeff  a000      ldy #$00
af01  d17a      cmp ($7a),y
af03  d003      bne $af08
af05  4c7300    jmp $0073
```

1. lista

```
aefa  a928      lda #$28
aefc  2ca92c    bit $2ca9
aeff  a000      ldy #$00
af01  d17a      cmp ($7a),y
af03  d003      bne $af08
af05  4c7300    jmp $0073
```

2. lista

3. lista

```
aefd  a92c      lda #$2c
aeff  a000      ldy #$00
af01  d17a      cmp ($7a),y
af03  d003      bne $af08
af05  4c7300    jmp $0073
```

Z80 programok haladóknak Spectrumra és Primóra

4. A Paint rutin

Grafikus programokban gyakran előfordul, hogy képernyőn levő zárt alakzatot kell befesteni. Erre, mint látni fogjuk, nekünk is szükségünk lesz a manószerkesztő programunkban.

A PAINT nevű program, amely az *1. listán* olvasható, erre alkalmas. Igen gyorsan működik, tetszőleges konkáv alakzatot is befest, valamint inverz festést (kiüritést) is tud. A program a *2. listán* szereplő változtatásokkal Primón is működik. Spectrum-tulajdonosok ügyeljenek arra, hogy a program a szintárolló tartalmát nem veszi figyelembe! A festés sebessége körülbelül 10 000 pont másodpercenként.

A program használata

A feltöltendő alakzat belsejében jelöljünk ki egy tetszőleges pozíciót! Koordinátáit a BC regiszterpárba töltsük (B=y, C=x). Ezen a pozíción helyezzünk el egy pontot, ha feltölteni kívánjuk, és legyen üres, ha kiüríteni szeretnénk az alakzatot. A CALL PAINT utasítással futtassuk a programot. A rutin az összes regiszter tartalmát megőrzi.

Spectrum BASIC-ből is hívható a program, ekkor az indítási cím PROBA, a kezdőpontot pedig egyszerűen az utolsó PLOT vagy PLOT INVERSE 1, utasítás jelöli ki, a program a megfelelő rendszerváltozóban megtalálja a koordinátát. Primósok a BASIC CALL utasítással átadhatják a koordinátaadatokat, a PROBA rutint pedig ennek megfelelően írják meg.

A feltöltés elve

Az *1. ábrán* egy konkáv alakzat látható, amelynek belsejében még egy sziget is van. A szigetet persze nem kell feltölteni. Az indulási pontot X jellel jelöltem meg. Ezenkívül tetszőleges belső pontok is előfordulnak, amelyek a feltöltést nem zavarják. A feltöltés vízszintes sorok mentén halad.

Most az „egy sor befestése" művelet következik. Megkerestük a sor elejét. Az 1. ábrán ez az 1 jelű pozíció. Innét kezdve balról jobbra haladva feltöltjük a sort addig, amíg pontot nem találunk. Az 1. jelű

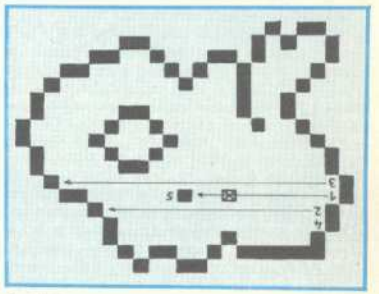
```
*L-
*C-

; ********************
; *                  *
; *    PAINT  9.2    *
; *                  *
; *  (c)1986 UHI-sw  *
; *                  *
; ********************
;
        ORG  $F000
PAINT   PUSH AF
        PUSH BC
        PUSH DE
        PUSH HL
        CALL TUBINI
        CALL PIXADD
        CALL MASKG
        CALL MODE
        CALL PUTTUB
PAINT1  CALL PLINE
        JR   NZ,PAINT1
        POP  HL
        POP  DE
        POP  BC
        POP  AF
        RET

MODE    LD   C,0
        CALL POINT
        RES  3,C
        RET  Z
        SET  3,C
        RET

TUBINI  LD   HL,FIFO
        LD   (READER),HL
        LD   (WRITER),HL
        RET

GETTUB  LD   HL,(READER)
        LD   DE,(WRITER)
        CALL CP16
        RET  Z
        INC  HL
        LD   E,(HL)
        INC  HL
        LD   D,(HL)
        INC  HL
        LD   B,(HL)
        PUSH DE
        LD   DE,FIFOE
        CALL CP16
        JR   NZ,GET1
        LD   HL,FIFO
GET1    LD   (READER),HL
        POP  HL
        XOR  A
        INC  A  ;NZ kell!
        RET

PUTTUB  EX   DE,HL
        PUSH DE
        LD   HL,(WRITER)
        INC  HL
        LD   (HL),E
        INC  HL
        LD   (HL),D
        INC  HL
        LD   (HL),B
        LD   DE,FIFOE
        CALL CP16
        JR   NZ,PUT1
        LD   HL,FIFO
PUT1    LD   DE,(READER)
        CALL CP16
        JR   Z,PUT2
        LD   (WRITER),HL
PUT2    POP  HL
        RET

PLINE   CALL GETTUB
        RET  Z
        CALL SLINE
        SET  0,C
        SET  1,C
PLINE1  LD   A,(HL)
        CALL POINT8
        JR   Z,PLINE2
PLINE3  CALL POINT
        RET  NZ
        CALL PLOT
        CALL BOT1
        CALL TOP1
        RRC  B
        JR   NC,PLINE3
        JR   PLINE5
PLINE2  XOR  A
        BIT  3,C
        JR   NZ,PLINE4
        CPL
PLINE4  LD   (HL),A
        CALL TOP8
        CALL BOT8
PLINE5  INC  HL
        LD   A,L
        AND  $1F
        JP   NZ,PLINE1
        INC  A   ;nz !
        RET

POINT   LD   A,(HL)
        AND  B
        BIT  3,C
        JR   NZ,P01
        OR   A
        RET
P01     XOR  B
        RET

PLOT    LD   A,(HL)
        OR   B
        BIT  3,C
        JR   Z,PL1
        XOR  B
PL1     LD   (HL),A
        RET

BOT1    PUSH HL
        CALL HLDOWN
        LD   DE,SCREND
        CALL CP16
        JR   NC,BOT12
        CALL BOTB
BOT12   POP  HL
        RET

BOTB    CALL POINT
        JR   Z,BOTB1
        SET  0,C
        RET
BOTB1   BIT  0,C
        RET  Z
        CALL PUTTUB
        RES  0,C
        RET

TOP1    PUSH HL
        CALL HLUP
        LD   DE,SCREEN
        CALL CP16
        JR   C,TOP12
        CALL TOPB
TOP12   POP  HL
        RET

TOPB    CALL POINT
        JR   Z,TOPB1
        SET  1,C
        RET
TOPB1   BIT  1,C
        RET  Z
        CALL PUTTUB
        RES  1,C
        RET

HLDOWN  INC  H
        LD   A,H
        AND  7
        RET  NZ
        PUSH DE
        LD   DE,-$07E0
        ADD  HL,DE
        POP  DE
        LD   A,H
        AND  7
        RET  Z
        LD   A,H
        ADD  A,7
        LD   H,A
```

```
181             RET
182
183 HLUP    LD      A,H
184             AND     7
185             JR      Z,HLUP1
186             DEC     H
187             RET
188 HLUP1   LD      A,L

189             AND     $E0
190             JR      Z,HLUP2
191             PUSH    DE
192             LD      DE,$06E0
193             ADD     HL,DE
194             POP     DE
195             RET
196 HLUP2   LD      A,L
197             SUB     32
198             LD      L,A
199             DEC     H
200             RET
201
202 SLINE   CALL    LEFTB
203             JR      C,SLINE1
204             RRC     B
205             RET
206 SLINE1  RRC     B
207 SLINE2  LD      A,L
208             AND     $1F
209             RET     Z
210             DEC     HL
211             LD      A,(HL)
212             CALL    POINTB
213             JR      Z,SLINE2
214             RLC     B
215             CALL    LEFTB1
216             RRC     B
217             RET     NC
218             INC     HL
219             RET
220
221 LEFTB   RLC     B
222             RET     C
223 LEFTB1  CALL    POINT
224             JR      Z,LEFTB
225             OR      A
226             RET
227
228 CP16    LD      A,H
229             CP      D
230             RET     NZ
231             LD      A,L
232             CP      E
233             RET
234
235 BOT8    PUSH    HL
236             CALL    HLDOWN
237             LD      DE,SCREN
238             CALL    CP16
239             JR      NC,BOT84
240             LD      A,(HL)
241             OR      A
242             JR      Z,BOT81
243             CPL
244             JR      NZ,BOT82
245 BOT81   CALL    BOTB
246             POP     HL
247             RET
248 BOT82   CALL    BOTB

249             RRC     B
250             JP      NC,BOT82
251 BOT84   POP     HL
252             RET
253
254 TOP8    PUSH    HL
255             CALL    HLUP
256             LD      DE,SCREEN
257             CALL    CP16
258             JR      C,TOP84
259             LD      A,(HL)
260             OR      A
261             JR      Z,TOP81
262             CPL
263             JR      NZ,TOP82
264 TOP81   CALL    TOPB
265             POP     HL
266             RET
267 TOP82   CALL    TOPB
268             RRC     B
269             JP      NC,TOP82
270 TOP84   POP     HL
271             RET
272
273 POINTB  BIT     3,C
274             JR      Z,P081
275             CPL
276 P081    OR      A
277             RET
278
279 MASKG   LD      B,$01
280             INC     A
281 MG1     RRC     B
282             DEC     A
283             JP      NZ,MG1
284             RET
285
286 *L+
287 FIFO    EQU     $-1
288             DEFS    400*3
289 FIFOE   EQU     $-1
290 READER  DEFS    2
291 WRITER  DEFS    2
292 PIXADD  EQU     $22AA
293 SCREEN  EQU     $4000
294 SCREND  EQU     $57FF
295
296 PROBA   LD      BC,($5C7D)
297             JP      PAINT
298             ENT     PROBA
```

1. lista

2. lista

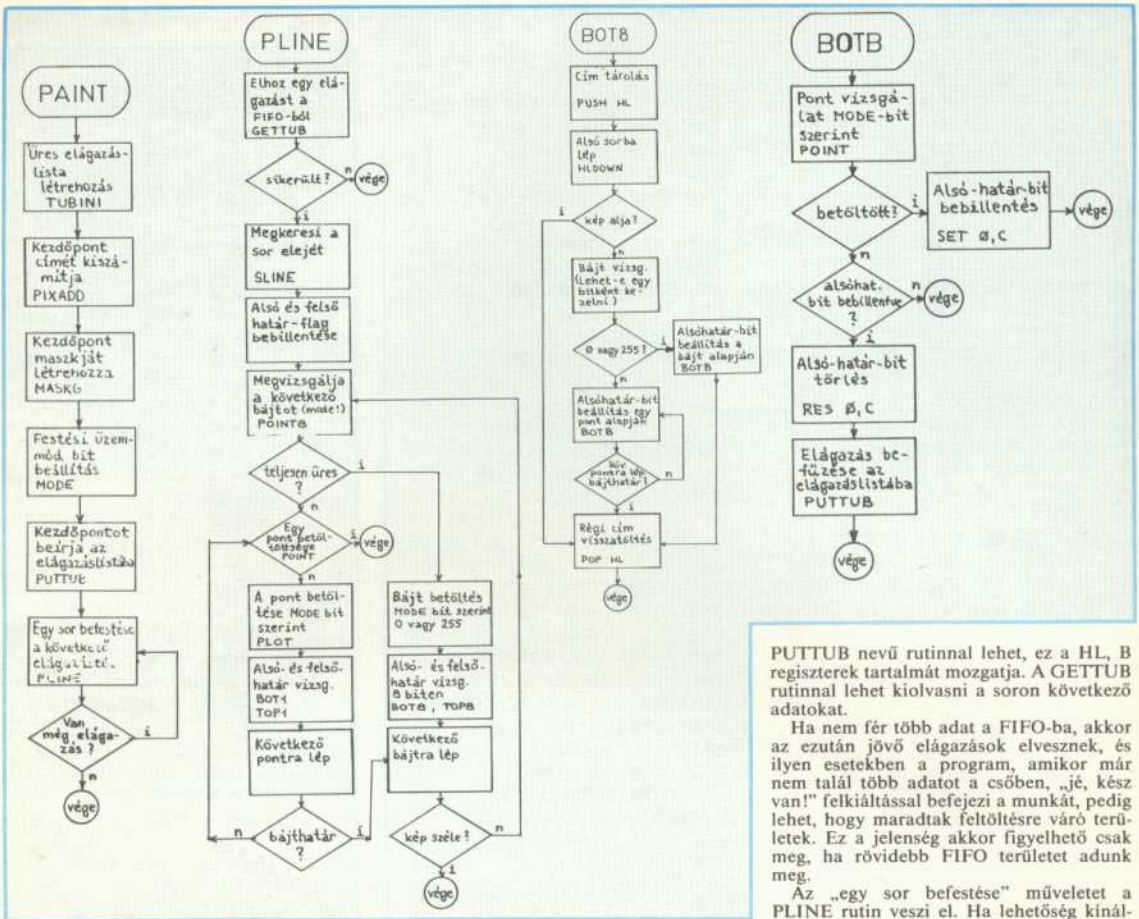
```
SCREEN  EQU     ...
SCREND  EQU     SCREEN+$17FF
;-----------------------------
PIXADD  XOR     A
        SRA     B
        RR      C
        RRA
        SRA     B
        RR      C
        RRA
        SRA     B
        RR      C
        RLA
        RLA
        RLA
        LD      HL,SCREEN
        ADD     HL,BC
        RET
;-----------------------------
HLUP    PUSH    BC
        XOR     A
        LD      BC,32
        SBC     HL,BC
        POP     BC
        RET
;-----------------------------
HLDOWN  PUSH    BC
        LD      BC,32
        ADD     HL,BC
        POP     BC
        RET
;-----------------------------
```



1. ábra

sor feltöltésekor éppen az egyik belső pontig jutunk el. Töltés közben figyelni kell, hogy nem bukkanunk-e elágazásra, külön a felette levő sorban, külön alul. Az elágazásokat megjegyezzük, vagyis egy elágazáslistába tesszük. Példánkban az 1 kezdetű sor feltöltésekor két elágazást lelünk: a 2 és 3 jelűeket. Ennyiből áll az „egy sor befestése".

Nézzük meg az elágazáslistát. Ha van még feltöltésre várakozó elágazás, akkor alkalmazzuk rá az „egy sor feltöltése" műveletet, és vegyük ki a listából. Ha az elágazáslista kiürül, akkor az alakzat teljesen be van festve. (Aki nem hiszi, CALL-jon utána . . .)



Figyeljük meg: előfordul, hogy egy elágazást kétszer tárol el ez az algoritmus. Példánkban a 2-es és 3-as kezdetű sorok feltöltésekor egyaránt elágazásként szerepel az 5 jelű pozició. Nem éri meg minden elágazás esetén végignézni a listát, hogy szerepelt-e már. Nagyon lassítja a működést. Az általam választott megoldás szebb: az elágazásokat FIFO szervezésű tárolóba teszem. A FIFO (first in, first out, de mondják tube-nak, vagyis csőnek is) egy olyan adatszervezést jelent, hogy az új elágazás mindig a lista végére kerül, a lista kiolvasása (a FIFO ürítése) pedig az elején történik. Tapasztalatom szerint így hamar kiürülnek a duplán tárolt elágazások, illetve kevesebb keletkezik, mintha egyszerűen és kézenfekvően vermet (stacket) használnék.

A program működése

A program megírásakor elsősorban a nagy sebességre törekedtem. Ez olyan fogások alkalmazásával járt, amelyek a program áttekinthetőségét rontották. A program csak a *2. ábrán* látható részletes folyamatábra segítségével érthető. A nagybetűs nevek a rutinok címkéi. Nem szerepel az ábrán minden rutin, csak a bonyolultabbak.

Az egyes pontkoordináták helyett mindenütt képernyőcím és maszk jelöli ki a pozíciót. A maszk egy olyan adat, amely a képernyőbájton belül jelöl ki egy pontot. Tehát a maszknak pontosan egy darab egyes bitje van.

A FIFO 400 rekeszes, vagyis 1200 bájt hosszú. Ez bőven elegendő. (Nekem még nem sikerült olyan ábrát rajzolnom, amiben egyszerre ilyen sok elágazás lett volna.) Paint rutinok tesztelésére tapasztalatom szerint egy olyan ábra alkalmas, amelynek minden második sorának minden második pozícióján van egy pont. Ugyanis ez generálja a legtöbb elágazást. Sok Paint program egyszerűen „kifekszik" ettől az ábrától. Egy korábbi változata a Paint programnak vermet használt, nem FIFO-t. Ezt a tesztábrát a szabad 40 kbájt memória igénybevételével sem tudta feltölteni! A FIFO-ban a következő üres helyre mutat a WRITER, a legöregebb adatra pedig a READER nevű mutató. A FIFO-ba írni a PUTTUB nevű rutinnal lehet, ez a HL, B regiszterek tartalmát mozgatja. A GETTUB rutinnal lehet kiolvasni a soron következő adatokat.

Ha nem fér több adat a FIFO-ba, akkor az ezután jövő elágazások elvesznek, és ilyen esetekben a program, amikor már nem talál több adatot a csőben, „jé, kész van!" felkiáltással befejezi a munkát, pedig lehet, hogy maradtak feltöltésre váró területek. Ez a jelenség akkor figyelhető csak meg, ha rövidebb FIFO területet adunk meg.

Az „egy sor befestése" műveletet a PLINE rutin veszi el. Ha lehetőség kínálkozik, együtt kezel egy teljes bájtot. Vagyis, ha a következő feltöltendő bájt 0, akkor oda 255-öt tölt, és megy a következő bájtra. Ez a fogás nagyon meggyorsítja a program működését.

Az alábbi rutinokat érdemes külön megnézni, mert más programokban is jól használhatók.

CP16: Összehasonlítja a HL és DE tartalmát. A flasek ugyanúgy állnak be, mint amikor az akkumulátorral hasonlítunk.
HLDOWN, HLUP: Egy képernyőbájt alatti, illetve feletti bájt címét adják meg. Ki- és bemenő paraméterünk a HL regiszterpár.
PUTTUB, GETTUB, TUBINI: A FIFO-t kezelő rutinok. A TUBINI inicializálja vagy teljesen kiüríti a csövet. A másik kettő használata szerepelt. Ha a FIFO betelik, vagyis nem fér bele több adat, akkor PUTTUB sem erőlködik, hanem Z flassel jelzi a hibát.

A következő részben ezt a programot kiegészítem úgy, hogy ne csak homogén befestést tudjon csinálni, hanem tetszőleges megadott mintázattal is tudjon zárt alakzatot tölteni.

UHERKOVICH PÉTER

Formázott listázás II.

A múlt hónapban közölt listázóprogram olyan nyomtatott programlistát állít elő, amelyről a program nagy megbízhatósággal reprodukálható. Ennek érdekében az eredeti listázóprogram szolgáltatásait kibővítettük. A bővítések eddig mind a programlista olvashatóságát növelték, csökkentve ezzel a program begépelésekor előforduló, a lista téves olvasásából származó hibák lehetőségét. Bármilyen figyelmesen olvassuk is el a programlistát, a begépeléskor még mindig elírhatunk valamit, ami aztán a program futását meghiúsíthatja.

Az adatok (jelen esetben a program) bevitelének általánosan elterjedt ellenőrzési módszere az, hogy a beírt adatokból egy *ellenőrző összeget* képzünk, majd ezt összehasonlítjuk az eredeti adatokból (programból) képzett ellenőrző összeggel. Ezt a feladatot sokféleképpen lehet megvalósítani.

Összegképzés és -kijelzés

A különféle megvalósítások két lényeges tulajdonságban térhetnek el egymástól: az ellenőrző összeg *képzésének módjában*, valamint az ellenőrző összeg *kijelzésének módjában*.

A legegyszerűbb módszerrel úgy képezhetünk ellenőrző összeget, ha minden egyes beírandó jel (betű, szám, jel) ASCII kódjait egymás után összeadjuk. Az így kapott összeget vagy közvetlenül használjuk ellenőrző összegként, vagy ebből az összegből képezünk egy újabb számot. Például vehetjük az így kapott számnak csak az alsó bájtját, vagy a szám bájtjait ismét összeadhatjuk, a számot ASCII kódként értelmezhetjük az ellenőrzéshez stb.

Az ellenőrző összeg kijelzése praktikusan a nyomtatás: az ellenőrizendő programot utólag egy másik program átvizsgálja, minden sorából képez egy ellenőrző összeget, majd ezeket táblázat formájában kinyomtatja. A táblázat első oszlopában a program sorszámai, a második oszlopban pedig az adott sorhoz tartozó ellenőrző összegek szerepelnek.

A módszer előnye, hogy ugyanaz a program használható akkor is, amikor a programlistához mellékelni akarjuk az ellenőrző táblázatot, és akkor is, amikor a program begépelése után a begépelt programról készítjük el ugyanazt a táblázatot. Hátránya, hogy az ellenőrző összegek nem programlistán, hanem egy külön táblázatban jelennek meg. A program begépelésekor a gépelési hibák kijelzése nem azonnali, hiszen csak a begépelést követő ellenőrzési fázisban lehetséges, amikor ismét elkészítjük az ellenőrző táblázatot, majd ezt soronként átnézzük és összehasonlítjuk az eredeti táblázattal.

```
SK 4000  REM-KONTROLL.SUM.-
XE 4010  REM----------------
CR 4020
QJ 4030 SUM =LN:POZ =0:ID =0:TXT$ = STR$
        (LN): GOSUB4230: RETURN
JQ 4040
FX 4050
HS 4060  REM-PRG.BYTE->SUM.-
JJ 4070  REM----------------
PB 4080
RJ 4090  IFID THEN4110
FQ 4100  IFA >127 THENTXT$ =KD$(A,0):
         GOTO4230
SD 4110 POZ =POZ +1:SUM =SUM +POZ *A:
         RETURN
FD 4120
KC 4130  REM-PRINT-KONTR.SUM
ME 4140  REM----------------
SF 4150
RH 4160 HI = INT(SUM /256):LO =SUM -256
        *HI:SUM =(HI ORLO) AND NOT(HI
        ANDLO)
EP 4170 OUT$ = MID$(KOD$,1 +(SUM AND15),
        1) + MID$(KOD$,1 +( INT(SUM /16)
        AND15),1) +"[SPC]"
XE 4180  GOTO2400
CK 4190
SX 4200  REM-TOKEN-V.-SORSZ.
PA 4210  REM----------------
XK 4220
GM 4230  FORI =2 TO LEN(TXT$):POZ =POZ +
        1:SUM =SUM +POZ * ASC( MID$(TXT$
        ,I,1)):NEXTI: RETURN
FP 4240
HG 4250  IF LEFT$(US$,1) = CHR$(143)
         THEN IF MID$(US$,2,1) ="$" THEN
         GOSUB2100: GOSUB2300
```

1. lista

Az újabb módszerek

Ezek más eljárást alkalmaznak. Külön program való az ellenőrző összeg képzésére akkor, amikor a program listáját nyomtatják, és egy másik program használatos a program begépelésekor. A két funkció különválasztása azzal a többlettel jár, hogy a nyomtatott programlistán minden sor tartalmazhatja az ellenőrző összeget is, fölöslegessé téve a külön táblázat készítését. Az igazi előny a program begépelésekor jelentkezik. Lehetőség van ugyanis arra, hogy minden egyes sor beírása után az adott sorból elkészítsük az ellenőrző összeget, ezt kijelezzük a képernyőre, majd azonnal összehasonlítsuk a programlistán szereplő összeggel. Vagyis a gépelési hibák már a program beírási fázisában felderíthetők.

Két programra van tehát szükségünk:

1. olyan listázóprogramra, amely minden utasításszort egy ellenőrző összeggel kiegészítve nyomtat ki,

2. olyan beviteli programra, amely minden beírt sorból képez egy ellenőrző összeget, és azt a képernyőre írja.

A két programnak természetesen ugyanazzal a módszerrel kell az ellenőrző összeget képeznie ahhoz, hogy a bevitel ellenőrzése eredményes lehessen. Olyan összegképzést kell tehát választanunk, hogy ne csak a saját programjainkról készíthessünk ellenőrző összeggel ellátott programlistát, hanem egyéb forrásból származó (nyomtatásban megjelent) programokat is ellenőrizve tudjunk begépelni.

A mi változatunk

Az összeg képzésére a COMPUTE!'s GAZETTE-ben 1987. január óta alkalmazott összegképzési módszert adaptáljuk. Mi nemcsak egyszerűen összeadjuk a program bájtjainak ASCII megfelelőit, hanem ennél kicsit többet is teszünk. Ennek a következő az oka. Ha például A=1234 helyett véletlenül A=1324-et írnánk, akkor a számjegyek összege (és az ASCII kódok összege is) ugyanaz maradna. Ezt a hibalehetőséget úgy küszöböljük ki, hogy az adott jel ASCII kódja helyett a jel ASCII kódjának és az adott jel utasításszorbeli pozíciójának szorzatát használjuk összegképzésre. Így ugyanaz a számjegy, betű vagy jel más-más értékkel növeli az ellenőrző összeg értékét, az utasításszorban elfoglalt helyétől függően.

Az ellenőrző összeg alapját maga a sorszám jelenti. Ehhez hozzáadjuk a sorszám jegyeiből az előbbiek szerint képzett összeget, majd az eljárást az utasítássor további jelein folytatjuk.

Külön figyeljük a BASIC utasításszavakat! Az utasításszavaknak nem az egybájtosan tárolt (tokenizált) formájából képezzük az ellenőrző összeget, mivel a program beírásakor nem a tokeneket, hanem magukat az utasításszavakat írjuk be. Beíráskor ezért nem lehet az utasításokat rövidített formájukkal helyettesíteni. Ha mégis a rövidített formához ragaszkodunk, akkor a következő a teendő. Miután beírtuk az utasításszorokat, ezeket LIST paranccsal a képernyőre íratjuk. Ekkor az utasításszavak ismét eredeti formájukban íródnak ki. Ha most soronként leütjük a RETURN-t, akkor a helyes ellenőrző összeget kapjuk.

Egy másik problémát okoz az utasításszorokban előforduló szóközök (SPACE) szerepe. Tudjuk, hogy ezek a program futását lassítják ugyan, de más hatásuk nincs. A nyomtatott programlisták jobb áttekinthetőségéhez szükségesek ezek a szóközök, de a program beírásakor el is hagyhatók. Az ellenőrző összeg képzésekor tehát ezekre a szóközökre nem kell figyelemmel lennünk. Nem hagyhatjuk el azonban az idézőjelek között elhelyezett szóközöket.

Az ellenőrző összeg képzésének utolsó lépése az összegként kapott szám átalakítása. A két bájtból álló szám egyes bájtjai között először KIZÁRÓ VAGY műveletet hajtunk végre. Így egybájtos számot kapunk, majd ennek az alsó és a felső négy-négy bitjét külön-külön két, 0—15 közé eső számként értelmezzük. E számokhoz egy-egy betűt rendelünk, és az ellenőrző összeg

helyett ezt a két betűt adjuk az ellenőrzéshez. A 0—15 számokhoz a program a következő betűket rendeli:

```
0 1 2 3 4 5 6 7
A B C D E F G H
8 9 10 11 12 13 14 15
J K M P Q R S X
```

Ezt a betűsorozatot tartalmazza a program 3030-as sorában a KO$ változó.

A listázóprogram bővítése

A múlt hónapban közölt listázóprogramban volt két szubrutinhívás, amelyek ott értelmetlennek tűnhettek. Ezt a két szubrutint most a 4000—4250 sorokban hozzáírhatjuk a programunkhoz. Ez a bővítés fogja elvégezni az ellenőrző összeg képzését és kiírását. A listán a sorszám előtti két szám maga az ellenőrző összeg. A program begépelésekor ezt nem kell beírni. Ha a most közölt KONTROLSUMMA nevű programot a LISTAZAS-2 program előtt bevisszük és futtatjuk, akkor ezt az ellenőrző összeget azonnal felhasználhatjuk a beírás ellenőrzésére.

Az eddig ismertetettek alapján az ellenőrző összeg képzését és kiírását könnyen megérthetjük. Segítségül szolgál az alábbi magyarázat is, amelyet az *1. lista* szerint lehet követni.

4030 Ellenőrző összeg képzése a sorszámból a SUM változóban. TXT$-ba a sorszámnak megfelelő karakterlánc kerül.
4090 Az idézőjel vizsgálata.
4100 BASIC alapszó beírása a TXT$ba.
4110 Ellenőrző összeg képzése egy programbájtból.
4230 Ellenőrző összeg képzése a TXT$-ban elhelyezett karakterláncból a BASIC alapszavak közé a listázóprogram által betett elválasztó szóközök figyelmen kívül hagyásával.
4160 Az ellenőrző összeg átalakítása kétbetűs kóddá. A BASIC-ben hiányzó KIZÁRÓ VAGY műveletet az (A OR B) AND NOT (A AND B) művelettel helyettesítettük.

2. lista

```
100  REMELLENORZOTTPROGRAMBEVITEL
110  REM(PROOFREADER)
120  REM------------------------------
130
1000 VEC = PEEK(772) + PEEK(773) *256:LO
     =43:HI =44
1010  PRINT"{CLR}{DOWN}{DOWN}{DOWN}":
     SPC(5);"ELLENORZOTT{SPC}PROGRAMBEV
     ITEL": PRINT SPC(15);
1020  IFVEC =42364 THEN PRINT"C-64"
1030  IFVEC =50556 THEN PRINT"VIC{SPC}20
     "
1040  IFVEC =35150 THENGRAPHIC CLR:
     PRINT"PLUS4{SPC}&{SPC}C-16"
1050  IFVEC =17165 THENLO =45:HI =46:GRA
     PHIC CLR: PRINT"C-128"
1060 SA =( PEEK(LO) + PEEK(HI) *256) +6:
     ADR =SA
1070  FORJ =0 TO166: READBYT: POKEADR,BY
     T:ADR =ADR +1:CHK =CHK +BYT: NEXT
1080  IFCHK < >20570 THEN PRINT"HIBA
     {SPC}AZ{SPC}ADAT{SPC}SOROKBAN": END
1090  FORJ =1 TO5: READRF,LF,HF:RS =SA +
     RF:HB = INT(RS /256):LB =RS -(HB *2
     56)
1100 CHK =CHK +RF +LF +HF: POKESA +LF,LB
     : POKESA +HF,HB: NEXT
1110  IFCHK < >22054 THEN PRINT"HIBAS
     {SPC}AZ{SPC}UTOLSO{SPC}ADATSOR,
     {SPC}A{SPC}PRG.UJRA{SPC}{SPC}{SPC}
     {SPC}{SPC}TOLTENDO": END
1120  POKESA +149, PEEK(772): POKESA +15
     0, PEEK(773)
1130  IFVEC =17165 THEN POKESA +14,22:
     POKESA +18,23: POKESA +29224, POKE
     SA +139,224
1140  PRINT CHR$(147); CHR$(17);"ELLENOR
     ZESRE{SPC}KESZ": SYS SA
1150  POKEHI, PEEK(HI) +1: POKEC PEEK(LO
     ) + PEEK(HI) *256) -1,0: NEW
1160  DATA120,169,73,141,4,3,169,3,141,5
     ,3
1170  DATA88,96,165,20,133,167,165,21,13
     3,168,169
1180  DATA0,141,0,255,162,31,181,199,157
     ,227,3
1190  DATA202,16,248,169,19,32,210,255,1
     69,18,32
1200  DATA210,255,160,0,132,180,132,176,
     136,230,180
1210  DATA200,185,0,2,240,46,201,34,209,
     8,72
1220  DATA165,176,73,255,133,176,104,72,
     201,32,208
1230  DATA7,165,176,208,3,104,208,226,10
     4,166,180
1240  DATA24,165,167,121,0,2,133,167,165
     ,168,105
1250  DATA0,133,169,202,208,239,240,202,
     165,167,69
1260  DATA168,72,41,15,168,185,211,3,32,
     210,255
1270  DATA104,74,74,74,74,168,185,211,3,
     32,210
1280  DATA255,162,31,189,227,3,149,199,2
     02,16,248
1290  DATA169,146,32,210,255,76,96,137,6
     5,66,67
1300  DATA69,69,70,71,72,74,75,77,80,81,
     82,83,88
1310  DATA13,2,7,167,31,32,151,116,117,1
     51,128,129,162,136,137
```

A múlt havi listázóprogramot tehát a most közölt kiegészítéssel együtt arra használhatjuk, hogy segítségével olyan jól olvasható programlistákat nyomtassunk, amelyek a program helyes begépelését egy utasításoronként alkalmazott ellenőrző összeg feltüntetésével segítik. A program begépelésekor az ellenőrző összeget a KONTROLSUMMA néven közölt program állítja elő.

A KONTROLSUMMA program

A program begépelésekor minden beírt utasításrobol egy ellenőrző összeget képez. Az ellenőrző összeg a képernyő bal felső sarkában jelenik meg egy két betűből álló jel formájában. Ennek a két betűnek egyeznie kell a listázóprogram által az utasítássor elé írt két betűvel. Ha nem így lenne, akkor a beírt utasításroban gépelési hiba van.

A gépi kódú program *(2. lista)* a BASIC terület elejére töltődik be, majd a BASIC elejét jelző mutatókat feljebb állítja. A program a Commodore gépcsalád bármely gépén átalakítás nélkül fut. Parancs módban, amikor a gép beviteli ciklusban van, a RETURN lenyomása után a program a beviteli pufferban elhelyezett adatokból képezi az ellenőrző összeget, majd azt a képernyő bal felső sarkába kiírja. Ezután a beviteli pufffer tartalmát ugyanúgy kezeli, mint máskor.

Megjegyzés. Az ellenőrző összeget előállító programok PROOFREADER néven terjedtek el. Az itt közölt program a COMPUTE!'s GAZETTE által alkalmazott módszer szerint képzi az ellenőrző összeget, ezért minden általuk közölt BASIC program beírását ellenőrizhetjük vele. A listázóprogramot ehhez a PROOFREADER-hez illesztve készítettem el azzal a céllal, hogy elsősorban a Mikromagazin számára készülő programlistákat bárki ellenőrző összegekkel ellátva küldhesse be, ezzel is csökkentve az igen gyakori bosszúságot okozó hibás programleírásokat.

ZSOM BÉLA

A szövegszerkesztő program (a DOC üzemmód)

A napi munka során rendszeres igény, hogy táblázatos jelentéseink, grafikonjaink szöveges részét ne kelljen más programmal megírni, és utána összeragasztgatni a részeket. A feladat felismerésének köszönhetően a LOTUS cég a SYMPHONY-ba a szövegszerkesztőt is beépítette, és így ebből az aspektusból is kényelmes eszközt adott a felhasználók kezébe. A SYMPHONY szövegszerkesztője sokoldalú és könnyen megtanulható. Talán az összes üzemmód közül ennek az elsajátítása a legegyszerűbb.

A szövegszerkesztő is a munkatábla kijelölt részén dolgozik, de csak egyetlen oszlopnyi területen tárolja a munkában levő szöveget. Ez a sorok szélességének korlátját eredményezi, vagyis egy szövegsor nem lehet hosszabb egy cella szélességénél: 240 karakternél.

Az üzemmód saját menüválasztéka a szorosan vett szövegszerkesztő funkciókat tartalmazza, míg az általánosabb feladatokat a SERVICE menüből lehet itt is elérni — mint például nyomtatás (teljes vagy részszöveg kijelölése, a margó beállítása, fej- és lábléc beállítása, oldalszámozás, lapméret meghatározása stb.), fájlműveletek (fájl mentése, visszatöltése, kombinálás más fájlokkal, lemez-tartalomjegyzék listázása, fájl törlése stb.), ablakkezelés (létrehozás, módosítás, törlés stb.).

Az a tény, hogy a szövegszerkesztő mindig egy oszlopot foglal a sorok tárolására (méghozzá csak a korlátozott ablak bal szélső oszlopát), lehetővé teszi, hogy egy időben párhuzamosan több dokumentumot írjunk, és a dokumentumok között egyetlen billentyű, az F6 WINDOW lenyomásával válthassunk. Az ablak méretétől és elhelyezkedésétől függően egy időben akár több szöveg vagy táblázat is látható a képernyőn, és a szövegrészek átvitele vagy másolása is könnyedén megvalósítható ezek között.

Tekintsük át a szövegszerkesztő parancsait, és ezen keresztül lehetőségeit. A parancsokat a MENÜ-ben találjuk, de a leggyakoribb parancsoknak funkcióbillentyűik is vannak (ezek a megfeleltetések a munka gyorsítását szolgálják).

Szövegek másolásakor (COPY), mozgatásakor (MOVE), illetve törlésekor (ERASE) ki kell jelölni a forrást: azt a tartományt, amelyről dolgozunk, ha a műveletet el akarjuk végezni. A tartománykezelés kicsit eltér a SHEET üzemmódban már leírtaktól (vö. a múlt havi számban a SYMPHONY első részével). Egyrészt a tartomány — ami itt karakterekből és nem cellákból áll — határának feloldása (ESC vagy BACKSPACE) után az újrarögzítést csak a TAB billentyűvel tehetjük meg

Mit tud

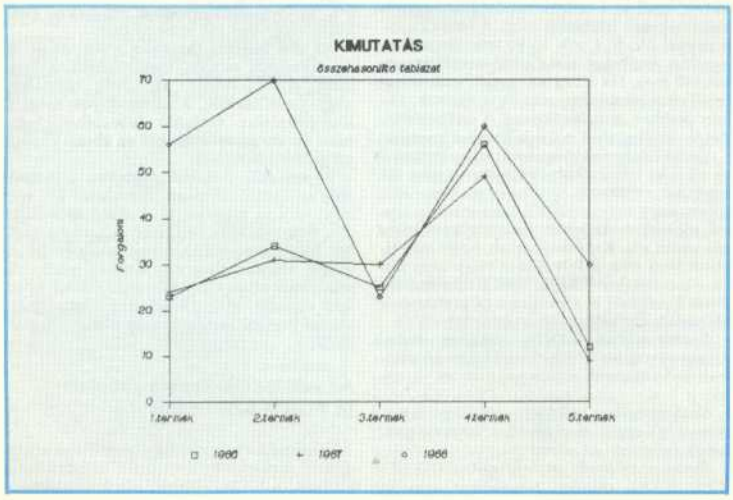
SYMPHONY? (II.)

```
     Most egy igen jó szövegszerkesztővel dolgozom. Ezzel a programmal
minden bekezdéshez külön formátumsort lehet készíteni.
                         L   T   T   T   T   T   T RE
                         Amit most írok, az csak ilyen
                         keskeny lesz, és kétoldali szöveg
                         illesztéssel, tehát ugynevezett
                         szövegkiegyenlítéssel jelenik meg

L   T   T   T   T   T   T   T   T   T   T   T   T   T   T TRE1
Egy bekezdésen belül elérhetjük egyetlen billentyű leütésére, hogy a bal
                margót átállítsuk. A képernyőn egy kis nyíl jelenik meg,
                amely természetesen a nyomtatott outputban nem látszik.
                Ebben a bekezdésben a sorok balra igazodnak.
A következő  bekezdésnél már  megint a  közvetlenül fölötte  lévő
formátumsor  beállításai (margók,  tabulátorok, szövegillesztés iránya,
írássűrűség stb) lesznek mérvadóak.
A betütipus a szövegben széles határok között megválasztható.
```

1. ábra

Éves eladások			
	1986	1987	1988
1.termék	23	24	56
2.termék	34	31	70
3.termék	25	30	23
4.termék	56	49	60
5.termék	12	9	30
összesen	2136	2130	2227

2. ábra



(ponttal nem lehet), másrészt a tartomány-kijelölés nemcsak a kurzormozgató billentyűkkel lehetséges, hanem egy karakter begépelésével, amely karakterig a tartományt a szövegben ki akarom jelölni.

Az anyagban meg lehet keresni konkrét szövegrész(eke)t (SEARCH), és könnyű valamit mással helyettesíteni (REPLACE). Az utóbbi szándéknál menüből előírhatjuk a helyettesítés módját, vagyis azt, hogy folyamatosan — a keresett szöveg minden előfordulásánál — vagy pedig szelektíven, csak az általunk kiválasztott esetekben változzék a szövegrész.

Egy tartalmában végleges szöveg újraformázásához szükséges parancs a szövegillesztés (JUSTIFY). Ezt bekezdésenként vagy a teljes szövegre előírhatjuk. Az illesztés iránya külön (több helyen is) beállítható.

A szöveg egészére, illetve egy-egy bekezdésre érvényes előírások a formátumvezérlő parancsban, a FORMAT-ban adhatók meg. Ezeket a beállításokat az ún. formátumsor tárolja. Egy ilyen sor addig érvényes, amíg a rendszer nem talál egy új formátumsort. Ez azt jelenti, hogy vagy az egész szövegnek azonos a formátuma (és ha ez az alapbeállítás, nem is kell vele foglalkoznunk), vagy bekezdésenként, bekezdéscsoportonként más-más formátumot kérhetünk. Bármelyik formátumsor elnevezhető, és ez bármikor — a nevére való hivatkozással is — megismételhető, majd szükség szerint módosítható.

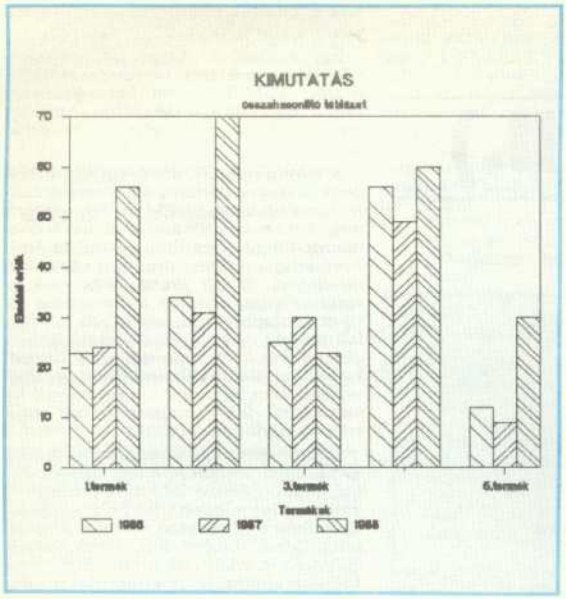
Mit tartalmaz egy formátumsor? A jobb és bal oldali margókat, a tabulátorpozíciókat, a szövegillesztés irányát (balra, jobbra, középre vagy mindkét irányba, ami a sorkiegyenlítést eredményezi), valamint a nyomtatott sorok közötti üres sorok számát (SPACE). A beállítások paraméterei bármikor visszaállíthatók az alapértékekre (RESET). *(1. ábra)* Ezek a szinte mindenre alkalmas alapbeállítások (DEFAULT), illetve széles választékuk az egész program során megkönnyíti a munkát és elsősorban a tanulást.

A program a SERVICE PRINT menüben kijelölt margóknak megfelelően a laptördelést automatikusan végzi, de sajnos a laphatárok nem láthatók a képernyőn. Lehetőség van egy billentyű leütésével lekérdezni a kurzor pozíciójában levő sor helyét a papíron (oldal- és sorszám), de ez a kiírás bármely művelet hatására eltűnik a képernyőről. A PAGE paranccsal kezdeményezhetjük, hogy a kurzor pozíciójától kezdődően, figyelmen kívül hagyva az adott oldal telítettségét, dobjon lapot, tehát kezdjen új oldalt.

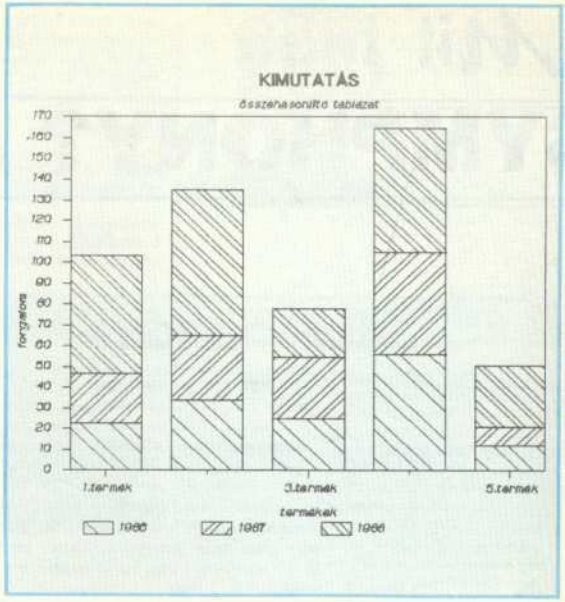
A DOC menütől függetlenül a funkcióbillentyűkkel további szolgáltatásokat érhetünk el. Egy-egy bekezdésre a bal margót egyetlen gomb lenyomásával állíthatjuk át. Szintén egy billentyűvel tudjuk a kurzor sorában levő szöveget középre helyezni, vagy egy másikkal a kurzor pozíciójától kezdve a szöveget a következő sorba átvinni.

Igen sok lehetőség kínálkozik arra, hogy a nyomtatási kép esztétikus legyen és jól kifejezze a tartalmat. A szövegen belül bárhol alkalmazhatunk az adott nyomtató által ismert tetszőleges betűformátumot, kicsinyítést, vastagítást, aláhúzást, indexelést stb.

Hogy ne tűnjünk elfogultnak, a szövegszerkesztő korábban említett hiányossága mellett még a következőket is megemlítjük: a szavak elválasztásához ez a program nem nyújt semmiféle segítséget, sőt a rugalmas kötőjelet sem kezeli. Sokszor okoz gondot, hogy ha a PRINT menüben kisebb szövegszélességet állítunk be a margókkal, mint amit a szövegszerkesztő formátumsorában megadunk, akkor a szöveg jobb oldala elvész anélkül, hogy erről bármiféle hibaüzenet tájékoztatna. Továbbá: a printervezérlő karakterek nem tüntethetők el a képernyőről, ami általában zavarólag hat a szöveg átnézésében, javításában.



3. ábra



4. ábra

A grafikai program (a GRAPH üzemmód)

Az emberek többsége képtelen nagy tömegű adat értékelésére, ha azt egy táblázatban tesszük elé. De ha egy olyan ábrával kiegészítve kapja kézhez az adatokat, amely szemlélteti a közöttük fennálló arányokat, már könnyen boldogul.

A SYMPHONY grafikus üzemmódja hatféle grafikont tud rajzolni. Ugyanarról a táblázatról akár mindegyik típussal készíthetünk ábrát, felváltva megjelenítve azokat, attól függően, hogy melyik tetszik vagy melyik tükrözi jobban a lényeget. Ha táblázatunkban egy vagy több adatot megváltoztatunk, ez azonnal a grafikon átrajzolását is magával hozza: így folyamatosan nyomon követhetjük a változások hatását. Az ábrákat a munkatábla elmentésével együtt tároljuk a rendszer, de ahhoz, hogy nyomtatón kirajzolhatók legyenek, külön ábrafájlba kell kerülniük, amelyet a SYMPHONY kiegészítő programja, a PRINTGRAPH fog értelmezni, és a kívánt nyomtatási formátumok beállítása után ki is tud nyomtatni. Beállítható a papír és az ábra mérete, az esetleges elforgatás szöge, az ábra feliratainak betűtípusa, színes nyomtató esetén a színek stb.

A grafikon adatainak, paramétereinek megadásához, ugyanúgy, mint a SYMPHONY-ban mindenhol, beállító táblák (setting sheetek) tartoznak. Minden grafikonhoz két tábla. A beállító táblákba az ábrákhoz tartozó specifikációkat kell megadni. Az ábrázolandó tartományok kijelölése után az alapbeállításokkal már megjelenithetjük az ábrát, feltéve, hogy grafikus ábrázolásra alkalmas monitorunk van. Ha nincs, akkor is készíthetünk ábrát, és a PRINTGRAPH programmal kinyomtathatjuk azt. A megjelenített ábra tartalmilag és esztétikailag tovább pontosítható — például ábrafeliratok, skálázás, magyarázó szövegek, rácsozat, viszonyítási érték — a beállítások további megadásával.

A GRAPH menü elérésére két lehetőségünk is van: vagy a SHEET menüből közvetve, vagy a GRAPH menüből közvetlenül. A feladat jellege és a munkamódszer sugallja, hogy melyiket kényelmesebb használni.

Hogyan készítünk el egy szép ábrát? Először is meg kell határozni, hogy a munkatábla vagy adatbázis mely tartományának adatait akarjuk az ábrában megjeleníteni. Hét tartomány kijelölésére van lehetőség, ezek közül az X tartománynak kitüntetett szerepe van: itt adhatjuk meg a vízszintes tengely feliratait — kivéve az xy típusú grafikont, ahol az x tengely valóságos numerikus értékeket jeleníthet meg. A többi tartománynál (A, B, C, D, E, F) tetszőleges numerikus értékeket tartalmazó területet adhatunk meg. Ha a tartományon belül véletlenül nem numerikus mezők is vannak, akkor ezekhez automatikusan Ø értéket rendel, és értelmetlen, csúnya ábrákat kapunk.

Ezután célszerű megnézni, hogy mennyire sikerült kifejező ábrát készíteni: nem túl zsúfolt-e, nincs-e egy kimagaslóan alacsony vagy magas értékű adat miatt a lépték rosszul megadva az automatikus skálázás során stb. Kipróbálhatjuk, hogy más típusú ábra nem kifejezőbb-e vagy nem adna-e esztétikusabb képet? Az alábbi hatféle ábrát készítheti el a program (a grafikonok alapadatait a *táblázat* tartalmazza).

Vonalgrafikon (LINE): minden tartományt egy vonal ábrázol, amelyhez ábrafeliratok járulhatnak a töréspontokban. *(2. ábra)*

Oszlopgrafikon (BAR): minden tartományt egy-egy oszlop ábrázol. Itt is megadhatók feliratok. *(3. ábra)*

Egymásra ültetett oszlopgrafikon (STACKED BAR): minden tartományt egy, az előző tartomány értékeire ráhelyezett oszlop ábrázol. *(4. ábra)*

Descartes-féle derékszögű koordinátarendszer (XY): az X tartomány az x tengelyen ábrázolandó numerikus értékeket tartalmazza.

Körgrafikon (PIE): csak az A tartomány értékeit ábrázolja, feltünteti az X tartomány címkéit, és kiszámítja az egyes körcikkek százalékos megoszlását. A B tartománynak kitüntetett szerepe van: itt határozzuk meg rendre, hogy melyik körcikket hogyan vonalkázza vagy színezze, és melyiket emelje ki a körből. *(5. ábra)*

Különbségi grafikon (HIGH-LOW-CLOSE-OPEN): az A és B tartományok különbségét egy függőleges vonallal ábrázolja, és a C és D tartományoknak megfelelő értékekhez egy-egy kis vonalat húz.

Az **első beállító táblában** a tartomány és az ábratípus mellett megadhatjuk a metszéspontok feliratát, az ábrák színezését vagy vonalkázását, a tartományok megkülönböztetésére szolgáló feliratokat, valamint az ábraelnevezéseket az ábrák közötti megkülönböztetésre.

A **második beállító táblában** adhatjuk meg az ábrák feliratait (ábracímet két sorban, az x és y tengelyt), a skálázás módját stb. Sok-sok más beállítási lehetőség mellett figyelemre méltó, hogy oszlopgrafikon esetén kijelölhetünk egy értéket, amely viszonyítási alapul szolgál: attól fölfelé és lefelé rajzolja az oszlopokat a viszonyítási alapul szolgáló értéktől való eltérés függvényében.

Az adatbázis-kezelő program (a FORM üzemmód)

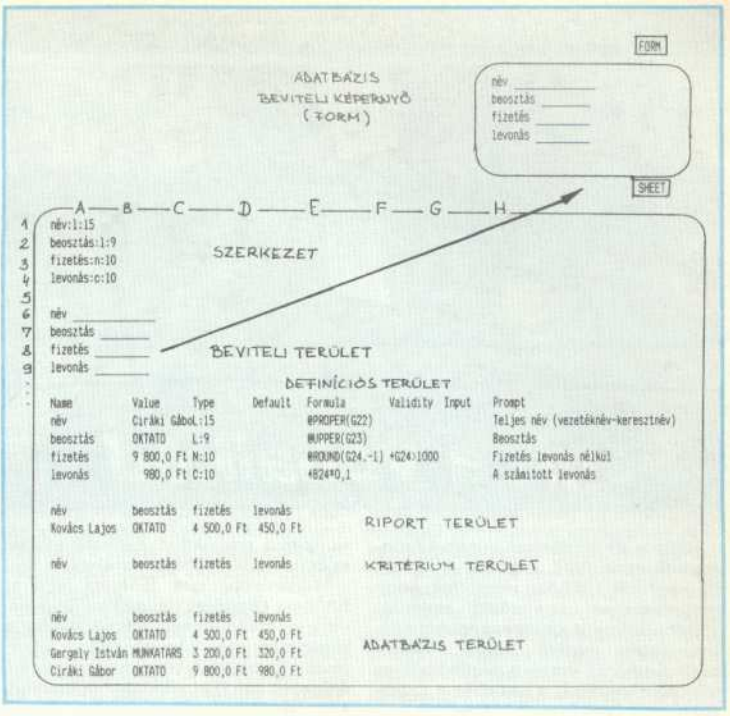
A számítástechnika legelterjedtebb alkalmazási területei közé tartozik az adatbázis-kezelés. A SYMPHONY adatbázis-kezelő-

je a dBASE III adatbázis-kezelő programnál megszokott relációs adatbázis-struktúrát használja, és lényegében azokat a lehetőségeket is biztosítja. Egy adatállomány egy táblázat, amelynek sorait rekordoknak nevezzük, és amely az egy egyedhez tartozó információkat tartalmazza (személyi adatok, raktárban az egy tételhez tartozó adatok stb.), a rekordokon belül az azonos tulajdonságú dolgok ugyanabba az oszlopba kerülnek. Az oszlopokat mezőknek nevezzük (név, cím, telefon, tételszám, darabszám stb.). Figyelembe kell azonban venni, hogy a SYMPHONY adatbázis-kezelője egy munkatáblára épül, és ez a különbség jelentős eltérést jelent a dBASE III-hoz képest.

Egy tárolt adatbázist nagyon sokféleképpen lehet használni. Más és más lista készíthető belőle, a legkülönbözőbb kritériumok szerint rendezhetjük, kereshetünk benne, kivonatokat készíthetünk stb.

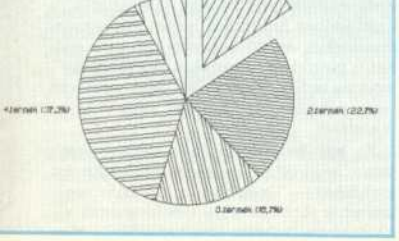
Egy adatbázis létrehozásakor a munkatáblákon (a SHEET üzemmódban) meg kell adni a mezők neveit, típusukat (szöveg, szám, dátum, idő, számított érték), valamint a mezőhosszakat, vagyis meg kell határozni az adatbázis struktúráját. Ezen adatok közlése után át kell kapcsolni a FORM (adatbázis-kezelő) üzemmódba. Ennek az üzemmódnak a menüjében kiadva az adatbázis-generálási parancsot, a program teljesen automatikusan, pillanatok alatt elkészít mindent, ami az adatbázissal való további munkához szükséges. Ez egyrészt a SHEET üzemmódban látható, másrészt a FORM-ban. A FORM üzemmódban tulajdonképpen a SHEET-nek egy kiragadott területén dolgozunk, amely lényegében egy „beviteli űrlap", amely megkönnyíti az adatbevitelt, és lehetővé teszi a SYMPHONY-nak az automatikus adatbázis-kezelést. *(6. ábra)*

Az adatbázis vezérlése a munkatáblának különböző, a generálás során létrejött tartományaiban történhet. A definíciós terület az adatbázis ún. „vezérlőterme". Itt az egyes mezőkre előírhatók ellenőrzések, átalakítások (kisbetű, nagybetű, kerekítés, számítási képlet stb.), megjelenítési formátumok, alapértelmezések. A riportterületen változatos nyomtatási formátumokat alakíthatunk ki az adatbázis listázásához. Ugyanahhoz az adatbázishoz több ilyet rendelhetünk hozzá, attól függően, hogy milyen célra készül a lista és mely adatokat akarjuk kilistázni. Az ún. kritériumtartományban jelölhetünk ki az adatbázisból egy-egy olyan részhalmazt, amely az itt megadott feltételnek (feltételeknek) eleget tesz. Ilyen területből is többet rendelhetünk



6. ábra

5. ábra



az adatállományhoz, biztosítva ezzel a felhasználás rugalmasságát. A kritériumok alapján kereshetünk az adatbázisban, listázhatunk stb.

Az adatbázis-kezeléshez szervesen kapcsolódnak az adatbázis-kezelő függvényei, amelyek a statisztikák készítésénél az adatbázis azon elemeit veszik csak figyelembe, amelyek az aktuális kritériumterületen lévő feltételeknek eleget tesznek. Ilyen függvények az átlagszámító, összegképző, mezőszám-meghatározó, minimális és maximális értéket megadó, valamint a szórást kiszámító függvények.

Az adatbázis elemeit három kulcs szerint lehet növekvő vagy csökkenő sorrendbe rendezni. A rendezés az adatok tényleges rendezését jelenti.

A SYMPHONY adatbázis-kezelésének nagy előnye, hogy az adatok bevitele és kezelése egyszerű és gyors. Az adatok beviteli formátuma és tárolási formátuma egymástól eléggé független. További előny a nyomtatási kép megtervezésének teljes szabadsága. Gondot okoz viszont a program felépítéséből adódó terjedelemkorlátozás.

Makrók a SYMPHONY-ban

A SYMPHONY programozható rendszer, amelynek programnyelve részint makroutasításokból áll (ezek a BASIC utasításokhoz hasonlóak), részint billentyűzési sorozatokból. Ez utóbbiak mindazokat a lehetőségeket tartalmazhatják, amelyekkel a SYMPHONY-t a billentyűzetről működtethetjük, vezérelhetjük. Programot írni akkor célszerű, ha egyrészt gyakran használt parancssorozatainkat akarjuk automatizálni (listázás, mentés stb.), másrészt, ha olyan felhasználó kezébe akarunk egy komplex rendszert adni, aki nem ismeri a SYMPHONY rejtelmeit. A munkatábla kezelésére szolgáló programot szintén a munkatáblán kell tárolni, bár lehetőség van programkönyvtár létrehozására is. A program utasításaiba az összes menüből kiadható parancs is beírható, de ez még csak egysoros programvégrehajtást eredményezne. A program attól lesz „igazi", hogy utasításai között megtaláljuk a feltételes és feltétel nélküli ugróutasításokat, a szubrutinhívást, az adatbeviteli utasításokat, a saját menükészítő parancsokat, lekérdezéseket stb.

A program utasításait a BASIC programnyelv utasításaihoz hasonlíthatnánk leginkább, de természetesen a munkatábla kezeléséhez szükséges szemléletet itt sem szabad elfelejteni. A makrókkal leírt program nehézkesen olvasható, teljesen strukturálatlan. De hogy mégse legyen olyan kényelmetlen, van egy tanuló üzemmódja, amikor az interaktív módon kiadott parancsokat „megtanulja" a rendszer, és a kijelölt programterületre lejegyzeteli a makróírás szabályai szerint.

Reméljük, hogy ez a rövid áttekintés elegendő ahhoz, hogy felhívja az olvasó figyelmét erre a kellemes és hatékony rendszerre. Érdemes beszerezni és a kezelését megtanulni, mivel annyi segítséget ad a mindennapi munkában, hogy kamatostól megtérül a befektetés. Hangsúlyozni szeretnénk, hogy már a program részleges ismerete is lehetővé teszi eredményes használatát.

GERŐ JUDIT—
DR. SZELEZSÁN JÁNOS

Segít a vonalkód

Noha a Skála Metrót a nagyközönségnek csak 1984. május 2-án adták át, arról már 1981-ben döntöttek, hogy nyugat-európai színvonalú, számítógépet alkalmazó kiskereskedelmi mintarendszert hívnak életre. Vagyis olyan áruházat, amely a későbbiekben mintául szolgálhat a hazai és a szocialista országok kereskedelmének.

Hat esztendeje számos nyugat-európai áruház megtekintése révén tanulmányozták, kutatták fel az alkalmazható megoldásokat. Egyértelművé vált, hogy a számítógép mellett az áruk nyomon követésére, értékesítésére a vonalkódos rendszer kiépítése az optimális. Mivel a hazai piacon sem gyártót, sem berendezést nem találtak — éppígy hiába kerestek szocialista piacon is —, maradt az, hogy az egész hardver konfigurációt tőkés termelőtől vásárolják. A lehetséges partnerek közül — mint a nagy hírű Hugin, NCR, Sweda, IBM, ADS Ander stb. — végül is az utóbbi mellett döntöttek, ugyanis ez szállított komplex rendszert, és árban is a legolcsóbb volt. A nyugatnémet gyártmányt a cég bécsi képviselőjétől a Skála Metró 7 millió 200 ezer schillingért vásárolta.

Ugyanabban az évben készültek fel a rendszer fogadására. Az áruházi áruforgalmi szakemberekkel összeállították a nagyvonalú, majd később a részletes rendszertervet. A felhasználási szoftvert saját fejlesztéssel készítették el. A programozási munkához már külső szakembereket is foglalkoztattak, és bérbe vették a Villért Honeywell gépét.

Mint Bencsik Sándor, a Skála Metró számítástechnikai osztályának vezetője az események sorolásakor külön is hangsúlyozta, nagy figyelmet fordítottak az előkészítésre, a gépek telepítésére, a felhasználói programcsomag megírására, a működéshez nélkülözhetetlen segédanyagokra, a kapcsolódó oktatásra, képzésre és megannyi olyan tennivalóra, amely látszólag nem tartozik szorosan a rendszerhez. Elkészült például az ügyviteli és működési szabályzat is.

Magyarország csak 1984 óta tagja az EAN-nek (Egységes Nemzetközi Kódrendszer), így korábban nem volt a termelőre, a készletezőre vagy a kereskedelemre egységesen alkalmazható cikkszámrendszer. Kidolgozták hát saját cikkszámrendszerüket, a vonalkódjuk alapját. A vonalkód formátuma szabványosan 13 vagy 18 pozíciós. Az EAN által előírttól az övék nem külsőben, hanem belső tartalmában tér el. Míg a szabvány országkóddal, termelői kóddal és cikkazonosító kóddal kezdődik, addig „házi használatban" ezeket nem igénylik, inkább áruházi osztályigazgatóság, osztálykód, cikkcsoportkód, cikkelem-azonosító és fogyasztói ár van feltüntetve a vonalkód

Thermotikett-készítés

pozícióin. Ezeket azután egy vörös (rubin) fénnyel működő olvasóceruza tapogatja le. Azért vörös fényű, mert a fehér fénnyel működő csak fekete-fehér kódot képes olvasni, színeset nem, viszont a fehér nem alkalmas masthermo papír olvasására, nálunk pedig éppen ez honos.

Hogyan olvas a kis ceruza? A pénztárgép kezelője végighúzza azt a tikettezett vonalkódos felületen. Azt, hogy jól leolvasta a készülék, hangjelzés igazolja vissza, s közben a kontroll- és a vevőszalagra kiíródik az áru azonosító száma, neve és fogyasztói ára. Ugyanakkor mindezt „nyilvántartásba" is veszik — de ne vágjunk a dolgok elébe.

A vonalkódos rendszert az áruház a teljes ruházati és iparcikkprofilnál alkalmazza, viszont máig sem használatos a hipermarketnél. Érdekesség, hogy a Skála Metróban fordítottan alakult a vonalkód bevezetése, mint az ötletszülő nyugati áruházakban. Az utóbbiaknál ugyanis éppen a hipermarketen és a kozmetikai áruknál vált be először, s csak később más profiloknál. Hogy nálunk miért e fordított helyzet? Az említett áruházakba az áruk már a vonalkód feltüntetésével érkeznek a termelőtől. A Skála Metróban viszont saját maguk tiketteznek, vagyis ott ragasztják fel valamennyi árura a vonalkódjelet. A számok bizonyítják: akkor célszerű és gazdaságos a rendszer működtetése, ha az áru 70-80 százalékán ott szerepel a vonalkód. Ezt ők elő is állítják, de az egyetlen rendszer üzemeltetőjétől nem várható el, hogy például több hektoliter tejet tikettezzen fel naponta. Hát ezért marad ki az élelmiszerrészleg a számítógépes rendszerből.

Végül is mi az előnye az egyedi és azonosítós vonalkódos rendszernek? A hagyományosan, azaz manuálisan működő áruház fogyasztói áras nyilvántartáson alapul. Vagyis csak egyetlen összegben tudják megadni, hogy mekkora a készlet, a forgalom stb. Azt például tudják, hogy a divatosztálynak 2 millió forint értékű áruja van, de hogy ez miből áll, azt már nem képesek megmondani. És azt sem, hogy a meglévőből mennyi fogyott, illetve mi érkezett, milyen a kereslet, melyik a lassan és a gyorsan fogyó cikk, miből van elfekvő készlet, miből célszerű árleszállítást, kiárusítást tartani. Ha leltároznak, akkor miután tudják, hogy — példánknál maradva — a divatosztály árukészlete 2 millió volt, akkor ennek meglétét vagy eladását keresik; de hogy esetleg melyik áruból van hiány, azt talán soha sem tudják meg, hiszen nem volt rá adatuk.

A vonalkódos nyomonkövetéssel minderről naprakészen kapnak tájékoztatást, s ezzel időben hozhatnak helyes áruforgalmi döntéseket. Az alacsonyabb készletek mellett is megfelelő választékot nyújthatnak — ugyanis tudják, hogy miből mennyijük van —, s pótolhatják a keresett vagy hiányzó árufféléket.

És bár hazánkban még nem követelmény, hogy effajta rendszer működjön egy áruházban — lévén hogy a manuális rendszerrel is többé-kevésbé elboldogulnak —, a változtatás példája ott lebeg a Skála Met-

Móra és a számítógép

Minden kezdet nehéz, de hogy ennyire? Azt nem is sejtették a Móra Ferenc Ifjúsági Könyvkiadóban, amikor életükben megjelent a számítógép. A technikai frontáttörésről, az indulás gondjairól, a tapasztalatszerzésekről Szakálos Mihály műszaki igazgatóhelyettessel beszélgettünk.

Tikett-tartó tasakok a méteráruosztályon

ró-sok előtt. Nevezetesen: az USA-ban már az 50-es évektől kényszerként jelentkezett a vonalkódos megoldás alkalmazása a kereskedelemben. Magyarországon a 90-es évektől kopogtat az igény a kereskedők ajtaján, s erre időben fel kell készülnie nemcsak a kereskedelemnek, de a termelő vállalatoknak is. És hogy mennyire így van ez, bizonyítja, hogy minderre országos kormányprogram született. A Skála Metró vállalkozása mintául szolgálhat másoknak.

De mit is tud a rendszer? Az árurendelés pillanatától az eladásig cikkelemesen, illetve cikkcsoportosan nyomon követi valamennyi cikk sorsát. A Honeywell 6/38-as központi gépen on-line módon, képernyőn jelenítik meg a rendelés-nyilvántartást, az áruérkeztetést, a fogyasztói ár kialakítását, a tikettezést, a készletezést és az áru mozgását a háttérraktár és az áruház, valamint a különböző osztályok között. Mindezt master-slave rendszerben oldják meg. Ez azt jelenti, hogy minden, ami a vonalkóddal a pénztárgépen történik, elsődlegesen a masteron (ez egy kisebb számítógép) keresztül floppyra kerül. Az ide begyűjtött adatokat zárás után, szintén a master segítségével feldolgozzák és kinyomtatják.

Azért nem mindjárt a központi számítógépbe gyűjtik az adatokat, hogy napközben ne terheljék. De azért sem, mert ha netán a gépteremben valamilyen hiba keletkezik, a rendszer akkor is üzemképes marad, hiszen egyéb gépekből mód van a cserére. A floppyra felírt adatok feldolgozásával a napi árufogyást leemelik a központi tételből. Így naprakészen tudják, hogy miből mennyi van, mennyi fogyott és milyen értékben; vagyis rendelkezésükre állnak azok az adatok, amelyek birtokában minden eddiginél rugalmasabbá válhat a kereskedelem.

KRASZNAI ÉVA

— A mikroszámítógép bevezetésekor kettős cél motivált bennünket. Egyrészt az elő- és az utókalkulációnál szakembergondjaink voltak, másrészt a nyomdák folytonos tarifaváltozásait lehetetlenség volt már mechanikus eszközökkel nyomon követnünk. Tehát a kényszer hozta, hogy alkalmazzuk a számítógépet. Meg nem tagadom, kíváncsiak is voltunk arra, hogy az annyit emlegetett — dicsért és szidott — számítógép valójában mit tud nálunk, a kiadói gyakorlatban. Elterjedt mindenesetre a hír, hogy a tarifák sokfélesége miatt képtelenség a kalkulációs munkához használható programot írni.

— *Milyen szempontok vezérelték önöket a géptípus kiválasztásánál?*

— A megfelelő számítógépet keresve, sokfelé érdeklődtünk. A tájékozódás nem volt könnyű, hiszen hazánkban a kiadói berkekben e területen nincs még kialakult gyakorlat. Végül is a Genie II típusú mikroszámítógépre voksoltunk.

— *Miért pont erre?*

— Mert a Kossuth Kiadónak volt már akkor egy ilyen gépe, amely ott bevált, sőt kalkulációs munkájukhoz programot is készítettek.

— *Mikortól datálódik a számítógép bevezetése?*

— Azt, hogy a Mórának számítógépet vásárolunk, még 1984-ben határoztuk el. Nem húzhattuk sokáig a beszerzést sem, mert a szükség nagy úr volt! Egy évvel később már át is vettük mikroszámítógépünket, a Genie II-t. És akkor az öröm helyett jött az eszmélés.

— *Nem volt jó a gép?*

— Fogalmunk sem volt arról, hogy mi fán terem egy számítógép. A berendezést nem ismertük, kezelni sem tudtuk, igaz, kapott egy szép, külön szobát . . .

Gyorsan továbbképeztük hát magunkat, hogy ha már egyszer beszereztük, ne árválkodjon tétlenül. Gépkezelőt és programozót is alkalmaztunk, a Kossuth Kiadótól pedig megvásároltuk a kalkulációs programcsomagot. Ezt ott öt nyomdára (Zrínyi, Kossuth, Athenaeum, Franklin és Szikra) készítették el. Tehát azokra, amelyekkel mi is kapcsolatban állunk. Kiderült, hogy a három további partnernyomdánkhoz nem készült program. De ha csak ez lett volna a baj! A gyakorlatban jöttek elő olyan dolgok, amelyekre tájékozottság hiányában nem gondolhattunk előre. Többek között az is problémát okozott, hogy míg a Kossuth Kiadó kevés illusztrációval, de sok szöveggel jelenteti meg a kiadványait, addig a Móra éppen fordítva, és ez más kalkulációt von maga után. A Kossuthtól kapott programot át kellett hát alakítani profilunkra. Vagy például a nyomdai kapcsolataink miatt bővítettük a nyomdákra való kiírásokat. És rengeteget kínlódtunk a nyomdánként változó tarifarendszer feldolgozásával. Majdnem igazuk lett a támadóknak. Végül is szakembereink elvágták a gordiuszi csomót. Mi tagadás, elég nehezen ment. Mert többek között a Kossuth programjában nem szerepel színbontás, a próbanyomat tarifájának feldolgozása, de nálunk ezek a tételek éppen hogy előkelő helyen állnak. Még egy probléma a sok közül: a Móra könyvei jobbára vékony gerincűek, két-három ív terjedelműek, a Kossuth könyvei viszont vastagok, jó néhány ív terjedelemben készülnek. Nem maradt más hátra, minthogy nekünk mindent — vagy majdnem mindent — helyi sajátosságainkra kellett adaptálnunk.

— *Ennyi gonddal járó programbevezetésnél megérte, hogy a Kossuthtól vásároltak?*

— Mi csak azt tudtuk, hogy nekik ugyanilyen gépük van, és hogy már írtak rá kalkulációs programcsomagot. Okkal gondoltuk, hogy a vásárlás után kezdhetjük is a próbakalkulálást.

— *Hogyan alakult tovább a Genie II bevezetése?*

— A számítógépes és a kalkulációs szakembereink feladatul kapták az adaptálást, és azt, hogy munkájukat 1986. március 31-ig készítsék el. Elképzelésünk ugyanis az volt, hogy az 1987-es kiadói tervünket már a számítógép közreműködésével állítjuk össze. A kis csapat nagyon igyekezett, becsülettel helytállt, megizzadva ugyan, de elkészült a különböző adaptálásokkal. A Mórában tavaly áprilistól honosítottuk meg a kalkulációban a számítógépes munkát.

— *Minden jó, ha jó a vége — a mikroszámítógép tehát bevált.*

— Méghozzá annyira, hogy több kiadó is érdeklődik a Mórában alkalmazott programcsomag iránt. A Gondolat például vásárolt már tőlünk programot.

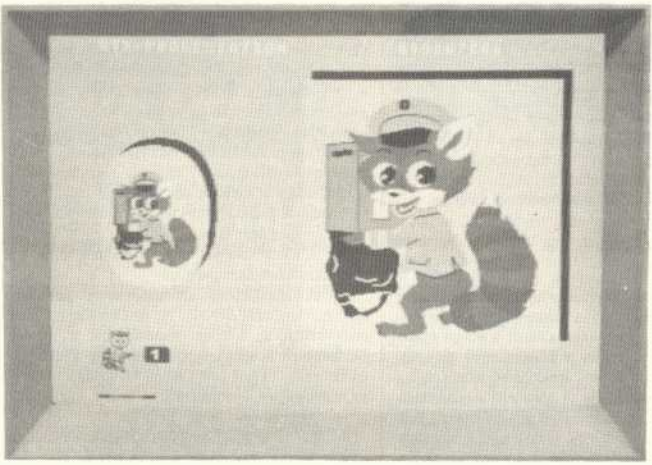
— *Nekik, ugye, zavartalan lesz az indulás?*

— Ott IBM gép van, tehát ők sem kerülhetik el az adaptálási munkát.

E rövid történetből is levonható valami tanulság. Annyi feltétlenül, hogy az átírások idejét, útját lerövidíthetnék a különböző kiadók, ha a jövőben a számítógép vásárlásakor gépparkjukat egységesítenék . . .

Miért nem stupid a MUPID? avagy: Milyen az intelligens videotex-dekóder?

Több nyugat-európai országban a mikroszámítógép-tulajdonosok a kapcsolt távbeszélő-hálózaton keresztül számítógépes adatbázisokból és egymástól is kérhetnek információt, sőt a mikrogépeiket működtető számtalan, hasznos-haszontalan programot is. Cikksorozatunkban a szomszédos Ausztriában honos és más országokban is alkalmazott MUPID videotex mikroszámítógépet mutatjuk be. Az írás aktualitását az adja, hogy a Magyar Posta várhatóan jövőre nyitja meg kísérleti videotex-szolgálatát — kezdetben csak közületi előfizetők számára —, amely a MUPID magyar változatát is ki fogja szolgálni. Nem célunk a videotexrendszerek ismertetése; ezekről kielégítő szakirodalom áll rendelkezésre. A sorozatban ennek a speciális mikrogépnek a korábbi, hasonló berendezésekhez képest nyújtott előnyeit, főbb műszaki jellemzőit, üzemmódjait vázoljuk, és bemutatjuk néhány, hazai szempontból előremutató alkalmazását is.



A MUPID-koncepció

Ki ne szeretné, ha kazettán vagy hajlékonylemezen garmadával sorakozna könyvespolcain több ezer oldalnyi friss érvényű lexikális és napi aktualitású információ, esetleg igény szerinti felhasználói program? „Ha van MUPID-od, ez a vágyad teljesül" — biztat az osztrák Maurer professzor, a MUPID konstruktőre. Elgondolása a MUPID létrehozásakor a következő volt.

A nyolcvanas évek elején léteztek már telefonhálózatokon keresztül elérhető, nagy mennyiségű információ lekérdezését és bizonyos ügyletek lebonyolítását lehetővé tevő videotexrendszerek (a továbbiakban vtx-rendszerek), melyek nagyon egyszerű és minden intelligenciát nélkülöző, olcsó lekérdező terminálokat szolgáltak ki. Az ilyen terminál lelke a dekóder, amely egyrészt fogadja és megjelenítésre alkalmassá teszi a távbeszélővonalon érkező soros digitális jeleket, másrészt előállítja az ábrázolható karaktereket, és legalább egy vtx-oldalnyi információt is képes tárolni. A vtx-felhasználót minden tevékenységében a vtx-központ támogatja. Így ha a vonali kapcsolat megszakad, a dekóder használhatatlanná válik. Az intelligensebb terminálok (általában a szerkesztő terminálok) többnyire ROM-ban lévő program alapján állnak a vtx-oldalakat helyi üzemmódban előállító információszolgáltatók rendelkezésére. Háttértárolót leginkább csak az elkészült oldalak összegyűjtésére használnak, melyekről az információ lemezen vagy vonalon juttatható el a vtx-központ atabázisába.

Az üzemelő vtx-rendszerek piacának növekedési üteme azonban különböző okokból jócskán elmaradt a várakozásoktól. Ez az üzemeltetőket a szolgáltatások minőségi és mennyiségi fejlesztésére sarkallta (Kanada: jobb minőségű, geometrikus megjelenítés és külső adatbázisok elérhetősége; Franciaország: elektronikus telefonkönyv, csak „külső" adatbázis-elérés, fejlett információkeresési módszer; Anglia: fotografikus megjelenítés).

Maurer professzor a korszerű információkereső rendszerek versenyében a vtx-alkalmazások bővítésének és a vtx-rendszerek továbbélésének lehetőségét az intelligens vtx-dekóderben látta. Ezen olyan számítástechnikai eszközt értett, amely a hagyományos vtx-funkciókon kívül digitális jelek helyi tárolására is alkalmas, és amelyet a felhasználó egy teljesen alfanumerikus billentyűzetről programozhat. Ezáltal az intelligens vtx-dekóder olcsó PC-vé válik, amely természetesen szót ért a vtx-központtal is.

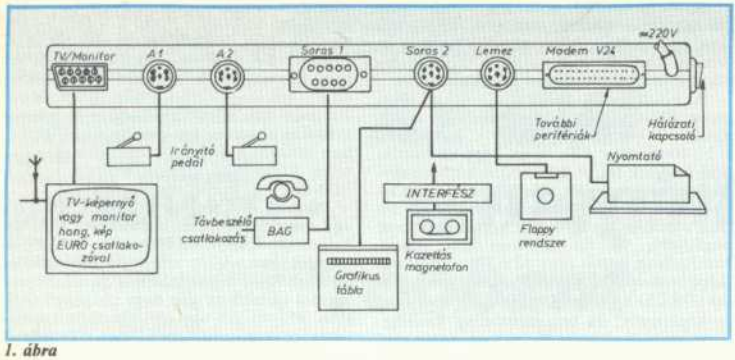
Ennek a megoldásnak két vonzó sajátossága is van. Egyrészt a felhasználó ún. teleprogramokat kérhet le a vtx-rendszerből és futtathat helyileg; ily módon potenciálisan hatalmas könyvtárat tudhat magáénak. Másrészt az intelligens vtx-dekóderét PC-ként, a vtx-rendszert saját programjainak és adatainak külső háttértárolójaként foghatja föl, azaz visszatöltheti ezeket a vtx-rendszerbe; szükségtelen a helyi háttértár (például mágneslemezes tár) beszerzése. És ami igen lényeges, a telefonvonalat csak a programok átvitelének idejére foglalja le, azok futtatása már helyi üzemmódban történik. Az ilyen dekóder a betölthető programok sokrétűségével válik — mint a neve is kifejezi (*M*ehrzweck *U*niversell *P*rogrammierbarer *I*ntelligenter *D*ecoder) — igazi többcélú, általánosan programozható intelligens dekóderré *(1. ábra)*. A dekódert, tartozékait és a működtető programok jó részét előállító grazi cég neve: MUPID Computer GmbH.

Mire jó a MUPID?

Nézzük át röviden, miért is vesz ma valaki mikroszámítógépet:

— mert szövegfeldolgozásra kell neki: leveleit, kéziratait, különféle számláit azon szeretné megírni és kiállítani. Ha a MUPID-hoz egy EPSON vagy Mannesmann Tally nyomtatót csatlakoztatunk, majd betöltünk egy MTED—80 vagy akár egy Wordstar programot, ez a kívánság rögtön teljesül;

— mert szabadidejét számítógépes játékokkal tölti (sakk, rejtvények, bridzs stb.). A MUPID-dal az osztrák vtx-központból mintegy 30 játékprogram hívható le anélkül, hogy magáért a programért fizetni kellene (a telefon-összeköttetés díját természetesen állni kell), de az ún. fizetős játékprogramoknak is tucatjai állnak rendelkezésre a legkülönbözőbb műfajokban;



1. ábra

— mert a különféle oktatóprogramokkal időt és pénzt kímélve kívánja magát továbbképezni. A MUPID igen korszerű oktatóprogramok fogadására alkalmas, melyeket akár lemezről, akár a vtx-központból lehet betölteni. Ezek száma most, 1987 nyarán már több száz;

2. ábra

60	2	3	4	5	6	7
0		0	@	P		p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	¤	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7	'	7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
A	*	:	J	Z	j	z
B	+	;	K	[	k	{
C	,	<	L	\	l	|
D	-	=	M	]	m	}
E	.	>	N	^	n	~
F	/	?	O	_	o	■

62	2	3	4	5	6	7
0		°	`	—	Ω	K
1	¡	±	´	®	Æ	æ
2	¢	²	^	©	Đ	đ
3	£	³	~	TM	ª	ð
4	$	x	¯	♪	Ħ	ħ
5	¥	µ	˘	J	I	ı
6	#	¶	˙	J	IJ	ij
7	§	·	¨	J	Ł	ĸ
8	¤	÷	¸	J	ł	ŋ
9	'	'	˚	J	Ø	ŉ
A	"	"	°	J	Œ	œ
B	«	»	¸	J	º	ß
C	←	¼	'	⅛	Þ	þ
D	↑	½	"	⅜	Ŧ	ŧ
E	→	¾	˛	⅝	Ŋ	ŋ
F	↓	¿	ˇ	⅞	ŉ	■

— mert ezzel véli kisebb-nagyobb számítási feladatait megkönnyíteni. A MUPID alkalmas a Multiplan és hozzá hasonló táblázatkezelő programok futtatására;

— mert jól mutatna egy katalógus saját könyv-, hanglemez- vagy ételrecept-gyűjteményeiről. A MUPID erre is megoldást kínál, hiszen futtathatók rajta az ismertebb dBASE fájlkezelő programok;

— mert szeretne programozni vagy azt megtanulni. A MUPID a megfelelő szoftverrel (akár a vtx-központból is lehívható-an) alkalmas BASIC, Turbo-Pascal, Z80 Assembly, sőt C és Modula—2 nyelveken való programozásra, és támogatja a teljesen kezdőket is;

— mert különleges vezérlésekre kívánja használni, mint például zenei effektusok előállítására, fényorgona vezérlésére stb. A MUPID természetesen ilyen esetekben is könnyen programozható;

— mert kézenfekvő, hogy a PC-tulajdonosok általában a nyilvános távbeszélő-hálózaton nyújtott szolgáltatásokat (mail-box,

Szabadon definiálható karakterek(DRCS)

DRC

3. ábra

4. ábra

CEPT Geometria (C2)

CEPT Alfmozaik DRCS (C0)

	C2	C0
DRCS karakterek száma	0	43
tárigény/Byte	132	1027
egy CEPT-oldal %-ában	6,9	53,5
betöltési idő (mp)	0	ca 8
előkészítési idő/perc	02	45
szerkesztési idő/perc	10	180
előállítási költség 10.-ATS/perc=összes ATS	120	2250

5. ábra

valósidejű levelezés) életük „peremfeltételeként" vehetik számításba. A MUPID a megfelelő programmal és akusztikus modemmel — más személyi számítógépekhez hasonlóan — erre is alkalmas.

És még mire?

A MUPID-dal megfelelő vtx-modem csatlakoztatása után az alábbi — például az osztrák vtx-központban nyújtott — szolgáltatások is elérhetők. (Az IBM PC-k, utánzataik és más mikroszámítógépek csak további hardverkiegészítéssel — általában vtx-dekóderkártya és vtx-szoftver bővítéssel — képesek ugyanerre.)

— Bármikor fellapozhat egy kb. 10 kötetes, folyamatosan aktualizált lexikont.

— Elektronikus úton — a dekóder billentyűzetéről begépelve — üzenetet küldhet más vtx-előfizetőknek, sőt olyanoknak is, akiknek nincs videotex-terminálјuk; és hasonló módon ő is kaphat üzenetet az elektronikus postaládájába, amelynek tartalmát csak hozzáférési jogosultságának igazolása után nézheti meg.

— Az ún. elektronikus bankon keresztül — a hozzáférési jogosultság igazolása után — bármikor lekérdezheti bank- vagy OTP-számláját, és így különböző pénzügyleteket hajthat végre (vásárlás átutalással akár távoli helyszínről is: teleshopping).

— Repülő- és vasúti jegyeket rendelhet, társasutazásra fizethet be, szállodákba és különböző kulturális rendezvényekre foglalhat helyet és azokról részletes ismertetőt kérhet.

— Speciális konvertálóprogram betöltésével együttműködésre alkalmas más vtx-rendszerekkel is: például az angol PRESTEL vagy az észak-amerikai NAPLPS vtx-rendszerekkel.

— A felsorolásból nem maradhat ki az itthon már ismert Képújsághoz hasonló informálódás lehetősége (aktuális hírek, időjárás, útviszonyok).

Miért vonzó?

Nem szóltunk még a dekóder kínálta grafikus lehetőségek bőséges választékáról. A MUPID által szélesebb körben is élvezhetővé válik a számítógépes grafika és a digitalizált képátvitel. Ezek forradalmian új vizuális lehetőségeket jelentenek a „fogyasztók" számára: az információ gyors lekérhetőségét és tálalását közvetlen analóg formában is.

A megjelenítési eljárások részletezése nélkül tekintsük át röviden a MUPID grafikus lehetőségeit. Megemlítjük, hogy a jelenlegi osztrák vtx-rendszer a CEPT-ajánlás C2 megjelenítési szintjét alkalmazza.

Az *alfamozaikos* megjelenítésnél a képernyőn 480 × 240 képpont a felbontás, a grafikus (megjelenítendő) karakterek 12 × 10-es vagy 12 × 12-es képpontmátrixból épülnek fel. A különböző karaktertáblázatokban definiált 355 ISO szabványú alfanumerikus és 151 mozaikkarakter *(2. ábra)*, valamint további, egy képen egyszerre legfeljebb 94 db, 5 × 6-os (16 színű), 6 × 10-es (4 színű) vagy 12 × 10-es (2 színű) képpontmátrixon szabadon meghatározható (DRCS) karakter *(3. ábra)* jelenhet meg a képernyőn, 24 (ergonómiailag kellemesebb megjelenítésnél 20) sorban, soronként 40 karakterhelyen. A soronkénti karakterszám nem vtx-oldal megjelenítése esetén legfeljebb 80.

Az *alfageometrikus* megjelenítési módnál a felbontás 320 × 240 képpont. A képernyő felületét egy kétdimenziós koordinátarendszerként foghatjuk föl. A vtx-oldal nem karakterről karakterre épül föl, mint az alfamozaikos eljárásnál, hanem a MUPID különböző, x, y koordinátáival megadott mértani alakzatokból (körív, négyszög, sokszög, csavart vonal stb.), ún. geometrikus elemekből szerkeszti meg a videotexképet *(4. ábra)*.

Mindkét megjelenítési módnál 16 előre meghatározott és 4096-ból tetszőlegesen kiválasztott másik 16 színárnyalat — tehát egy képen összesen 32 szín — használatára van lehetőség. A MUPID a beérkező adatfolyam élén álló vezérlőkarakterekből ismeri föl, hogy éppen milyen megjelenítési eljárás szerint kell az adatokat kezelnie. Az alfamozaikos karaktergenerátor és a geometrikus elemeket leíró szubrutinok ROM-ban vannak. A beépített, fejlettebb megjelenítési képességei alapján nevezik a MUPID-ot C2-es szintű vtx-terminálnak.

Digitális kép megjelenítésére a MUPID egy speciális program betöltésével válik alkalmassá. A vonalról érkező, az eredeti állóképet leíró analóg videojelből mintavételezéssel majd digitalizálással és adattömörítéssel kialakított igen nagy tárigényű digitális információt úgy alakítja monokromatikus fényképszerű képpé, hogy a képernyő egyes képpontjaihoz a programban előírt eljárás szerint maximálisan 16 különböző világosságérték közül a memóriájába betöltött, az adott képpontnak megfelelő címen levő videojelmintát rendeli *(5. ábra)*.

Mint az eddigiekből látható, az alkalmazási lehetőségek határtalan választéka tárul elénk. (Belső fedeleinken ízelítőül színesben mutatunk képeket minderről.) A távoktatástól a műholdról készült digitális meteorológiai felvételek továbbításáig, a bűnügyi azonosítási képek lekérhetőségétől a különböző névsorokban (telefon, telex, véradók stb.) való böngészésig — minden. (Folytatjuk.)

JURENKA OSZKÁR

Fogalmak

videotex-szolgálat: nemzetközi postai ajánlásban definiált olyan párbeszédes üzemű szolgálat, amely videotex-terminálok használói számára lehetővé teszi, hogy távközlési hálózaton keresztül, szabványosított elérési eljárások révén adatbázisokkal vegyék fel a kapcsolatot

videotext: a nyugatnémet ARD és ZDF televíziós társaságok képújság-szolgáltatásának márkaneve

Bildschirmtext (Btx): a videotex német elnevezése, egyben az NSZK és osztrák videotex-szolgáltatások márkaneve

videotex-szolgáltatások: alapvetően információlekérdezés, ügyletek lebonyolítása, üzenetközvetítés. Ide sorolható a tervezett telex- és telex-előfizetőkkel való kapcsolatfelvétel lehetősége, valamint a teleprogramok lehívhatósága

videotex-terminál: olyan berendezés, melynek segítségével a videotex-előfizető párbeszédes kapcsolatot folytathat a videotex-szolgálattal. Tipikus alkotóelemei a számjegyes vagy alfanumerikus billentyűzet, a megjelenítőegység és a dekóder

videotexközpont (vtx-központ): A videotex-szolgáltató (általában a posták) által üzemeltetett, a videotex-szolgálathoz hozzáférést nyújtó és a jogosultságot ellenőrző számítógép, amely támogatja az előfizetőket az adatbázis(ok) elérésében, díjaz, statisztikát készít

videotexkép (= videotex-aloldal): a videotex-terminálról egyetlen felhasználói paranccsal lehívható, a terminálon a teljes képernyőt vagy annak egy részét kitöltő információmennyiség

videotexoldal: a videotex-adatbázisban egy oldalazonosítóval (oldalszámmal) jelölt, de több, az angol ábécé kisbetűivel (tehát max. 26) egymástól megkülönböztetett aloldalakból álló információmennyiség

DRCS (Dynamically Redefinable Character Set): dinamikusan újradefiniálható karakterkészlet. Olyan speciális karaktertáblázat, amelynek elemeit (max. 94) adott felbontás mellett szabadon lehet meghatározni, majd a videotex-dekóderbe a videotexoldal karaktercímeit megelőzően kell betölteni

PRESTEL: az Európában elsőként megnyitott, az angol British Telecom által üzemeltetett nyilvános videotex-szolgáltatás márkaneve. Egyben az itt megvalósított megjelenítési szint neve is, melynek jellemzői tulajdonképpen a hazai képújság megjelenítési tulajdonságaival azonosak

CEPT megjelenítés: Conférence Européenne des Postes et des Télécommunications = az Európai Posta- és Távközlési Igazgatások Szövetsége által meghatározott videotex-szabványnak megfelelő megjelenítés, amely elvben felülről kompatibilis az Európa-szerte kialakult megjelenítési módokkal, így például a PRESTEL-lel is. Alfamozaikus megjelenítésorientált

NAPLPS: North American Presentation Level Protocol Syntax = az Észak-Amerikai Megjelenítési Szintű Protokoll Szintaxisa, az USA-ban és Kanadában elfogadott videotex-megjelenítési szabvány neve. Alfageometrikus megjelenítésorientált

Az IBM megint robbant

Az IBM maga sem számított rá, hogy a számítástechnikai piacon a PC/XT, később az AT gépei meghatározók lesznek. Az első IBM PC-k 1981 augusztusában jelentek meg, és olyan szabványt teremtettek, amelyhez azóta is kénytelen-kelletlen, de igazodnak a versenyben maradni akarók.

Ezt támasztja alá az a tény is, hogy 1985-re az IBM az amerikai piac 51 százalékát uralta, és a keresletet nem is tudta kielégíteni. Szokatlan jelenség ez az amerikai gyártásban.

Az IBM PC család viszonylag nyitott felépítése lehetővé tette, hogy más számítógépgyártók és szoftverfejlesztők hamarosan bővítőkártyákat és alkalmazói programokat kínáljanak a vevőknek. Az IBM-másolatok gyártása külön iparággá fejlődött. Ezek a másolatok — klónok — a piac mind nagyobb részét foglalják el, ami főleg az alacsonyabb áraiknak és a majdnem tökéletes kompatibilitásuknak tudható be. (Emlékeztetünk ez évi 5. számunk cikkére. — A szerk.) Az IBM — érthetően! — erőlködött, de nem tudta saját teremtménye másolatainak sikerét korlátozni.

A számítástechnikával foglalkozó szakemberek szinte egyöntetű véleménye, hogy az eredeti IBM gépekkel tökéletesen azonos tudású másolatokat nem sikerült kialakítani. Nagy vonzerő azonban, hogy ezek az utánzatok az eredeti gép árának mintegy 30 százalékáért megkaphatók, és nem is roszszak.

1986-ban először, az IBM kevesebb gépet értékesített, mint a másolatok gyártói. Ezen a hirtelen árcsökkentéssel sem tudtak javítani. Tudomásul kellett venniük, hogy lemaradtak a piaci részesedésben. Úgy tűnik azonban, hogy újítani tudnak. Tavasszal bejelentette a cég az új gépcsaládot, a PS/2-t.

Az új IBM gépcsalád négy alaptípusból áll: Model 30, 50, 60 és 80. Az új típusokat az IBM szerint nehezebb lesz utánozni, mert speciális anyaggal beburkolt, különleges chipeket használnak a gépek grafikájának vezérléséhez. Az új gépekbe szintén Intel gyártmányú mikroprocesszorokat építenek be, és az operációs rendszert a Microsoft készíti.

A PS/2 (Personal System) modellek összekapcsolhatók az IBM nagy és közepes számítógépeivel. Egyes típusai — a beharangozás szerint — multiprogramozásúak lesznek. Várhatóan a jövő év elejéig forgalomba kerülnek.

A CHIP magazin júniusi számában tesztelte a PS/2 Model 30-as gépet. Az erről szóló cikket ismertetjük.

A Model 30 a korábbi PC gépekkel kompatibilis. A gépcsalád legkisebb tagjára vetett első pillantással megállapítható, hogy a IBM alapjaiban újat fejlesztett ki. Az egészen könnyű, mindössze 8 kg súlyú gép kompakt, vagyis a billentyűzet és a képernyő egymáshoz van építve. A hagyományos személyi számítógépek egyenesen formátlannak tűnnek az új gép mellett *(1. kép)*.

Mindjárt megemlítünk egy műszaki változást: a hajlékonylemez-meghajtó 3,5"-os. Igaz, hogy ez megnehezíti az átjátszást a korábbi PC modellekkel, de a kisebb lemezre való áttérés már esedékes volt. Igény szerint 20 Mbájtos merevlemezegységgel kapható, és kulccsal elzárható kivitelben készül.

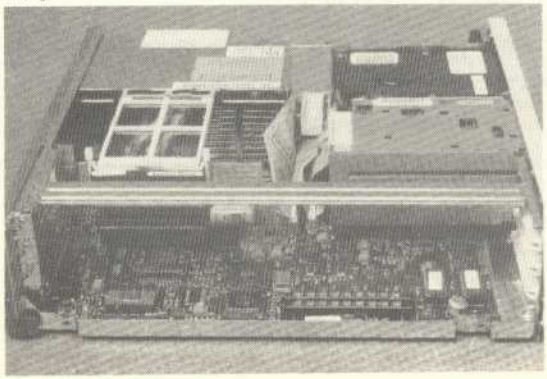
Vitatható a számítógép formatervezésének néhány megoldása. Elejének kialakítása sajátságos, szokatlan. A piros színű hálózati kapcsolót is itt helyezték el. További újdonságot mutat a hátoldal: a billentyűzethez és az „egér"-hez új csatlakozót alakítottak ki.

A gép alaposabb vizsgálatánál tapasztalhatjuk a további fejlesztést. Már a fedél leemelése is egyszerűbb, mint a PC-nél, mindössze négy csavart kell kilazítani. A fedél eltávolítása után megláthatjuk a legmodernebb technikai megoldást: a panel felszínére forrasztott alkotóelemeket (SMD = surface mounted devices) és alkalmazásorientált integrált áramköröket (ASIC = application-specific integrated circuits) *(2. kép)*. Megbízhatónak csak a három fő panel tűnik. Ezeket vízszintesen helyezték el, egyébként megegyeznek a korábbi IBM gépeknél megszokottakkal.

A gép belseje igen jó van elrendezve. A 640 kbájtos memóriachip két kicsi, átlósan felfelé álló hordozón helyezke-

1. kép

2. kép



dik el. Nem tudjuk, mi okból van erre szükség. Mindent, ami a számítógép normál működéséhez szükséges, a fő panelen egyesítettek: párhuzamos és soros interfészek, képernyővezérlő, lemezmeghajtó és merevlemez-vezérlő, valamint egy egér-interfész. Így a gép fő kártyája valóban bővíthető.

A PC/XT-től különbözően a Model 30-nak van még egy párhuzamos interfész-csatlakozója. Ez kétirányú, mind kimenetként, mind egy külső meghajtó részére szolgálhat. A megoldás hasonló ahhoz, ami a Toshiba Handhelds gépénél használatos, és ami a CHIP magazin szerint korszerűtlen.

Az új IBM számítógéphez négy különböző monitor kapható: egy fekete-fehér és három színes. Az IBM-től megszokottal szemben a monitor analógjel-erősítését a számítógépben helyezték el, így tulajdonképpen az végtelen sok színű telítettséget tud megjeleníteni. Valójában a képernyővezérlés a legnagyobb technikai változtatás a PC-hez képest.

Mindezek ellenére a gép grafikus ábrázolása és annak képességei kissé kiábrándítóak. Az ábrázolás 256 színből áll, 320 × 200 pontképpel. Ezenkívül áttérhetünk 640 × 480-as pontképhez is. Ez modulárisan érhető el. Egyéb tekintetben megmaradtak a hagyományos, normának tekinthető, szűkös grafikánál. Ezen a területen többet várhattunk volna: legalább az IBM saját, ún. EGA rendszeréhez hasonlót.

Látványosabb viszont a képábrázolás minősége. A képfrekvencia (a fénypontok rezgésének ideje) 70 Hz. A képernyő teljesen csillogásmentes, a feliratok kiemelhetők. A CHIP magazin két típust tesztelt: a színes közepes minőségűt és a fekete-fehéret. Mind a kettő 30,5 cm átmérőjű. A készülékek nehéznek tűnnek, de praktikus, mozgatható talpon helyezték el őket. A nem túl olcsó (2000 DM) színes monitor bántóan erős, ragyogó színekkel fénylik. A monochrom monitor szintén nem olcsó, és rádadásul még képtelítettsége is gyenge. Amint a használat során kiderült, 64 árnyalatot képes megjeleníteni. Néhány árnyékolás alig látható, és gyakran át kell állítani a monitort, illetve a gépet világosról sötétre, hogy a képet egyáltalán látni lehessen.

Az IBM-nek talán jobban kellett volna ügyelnie a tónusokra és a képernyő vezérlésére. A kimenetet ugyan praktikusan alakították ki, hiszen automatikusan a saját monitorállására küldi a gép a jeleket. A PS/2 semmilyen „kapcsolót" nem igényel, az üzemkezdetnél felismeri, hogy milyen elrendezésű.

A gép mindössze 70 wattot fogyaszt, kevesebbet, mint a PC. Ezt az alaktrészek műszaki fejlesztésével érték el: kevesebb energiát használnak.

A gép billentyűzete — mint már a korábbi típusoké is — csúcstechnológiát jelent. Az ötletes elrendezés (12 funkciógomb elkülönítése) teszi a billentyűzetet célszerűvé.

A Model 30 kompatibilis az IBM korábbi gépeivel, és a nagyszámítógépekkel is kapcsolatot teremt. Az új monitorvezérlő sem változtatja meg ezt.

A kompatibilitás az egyetlen eltérés, a lemezmeghajtó méretének eltérése miatt a tapasztalatok szerint csak 95 százalékos, ami kétségtelenül nem garancia a teljes azonosságra. A jelenlegi meghajtó, a soros MS—EGA (6.0 verzió) nem illeszkedik az új meghajtóhoz.

A mikroprocesszor a megszokott Intel 8088 helyett 8086 típusú, és 8 MHz-cel dolgozik. A soros és párhuzamos átvitel 19 200 baud.

Amennyiben ez összehasonlítható, a Model 30 összességében kétszer gyorsabb, mint a PC/XT; a képernyő-megjelenítés két és félszeres, a merevlemez-olvasás gyorsasága viszont csak az utóbbiénak a fele. Kétségtelen, hogy a mérésnél nem azonos normákat használtak.

Összegezve az elmondottakat: az IBM és a Microsoft a személyi számítógépek valóban új típusát dolgozta ki. A DOS, illetve MS—DOS 3.3 rendszere a hardverfeltételen túl javított az adatelérésen, és az operációs rendszer emberközelibbé vált. A teszt nem mutatott különösebb eltérést az MS—DOS 3.2-höz képest. Az MS—DOS 3.3 hibátlanul fut az IBM gépeken, és 2 kbájttal kevesebb helyet foglal.

Az új IBM gép áráról nem érdemes írni, mert ismerve a számítástechnikai ipart, napról napra változhat.

PINKE GYÖRGY

Szemüveg alatt a számítástechnika ártalmai

A cím megtévesztő. Nem a számítástechnika lehetséges ártalmairól beszélek, ha van ilyen egyáltalán, hanem a vele való foglalatosság fiziológiai összetevőiről.

Bizonyára sok szülő jelentkezne, hogy gyermekét nem érdeklik a hagyományos játékok: építőkocka, baba, papírépítő, de állandóan a képernyő előtt ül. Sőt, még jogos büszkeséggel mondja: az én gyerekem számítógéppel játszik! A gyerek ügyes programokat készít és néha még a felnőtteket is lekörözi tudásban. A szülő örül, hogy csemetéje értelmes elfoglaltságot talál, és nem gondol a képernyő veszélyeire. De milyen veszélyeket is rejt az állandó képernyőnézés?

Gyermekünk gyakran panaszkodik, hogy fáj a szeme, és nem lát jól. Mi lehet ennek az oka?

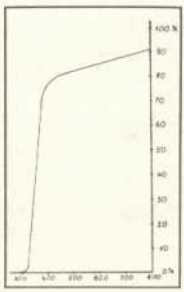
Dr. Vörösmarthy Dániel professzor a Szemüveg c. újságban (1987/1. sz.) cikket írt arról, hogy egyik legfontosabb érzékszervünket, a szemünket idegen – látható és láthatatlan – sugarak érik a monitor, a képernyő nézése következtében. Legártalmasabbak az ultraviola és az infravörös fénysugarak. E láthatatlan, de az idegháryára káros sugarak különösen a fiatalokra veszélyesek.

A CompuDrug Műszaki Fejlesztő Kisszövetkezet egyéb termékei mellett, mint a már ismert szoftverjei, a Flotix (lemezolvasófejtisztító), most az alkalmazók egészségvédelme céljából fejlesztette ki új találmányát, a védőszemüveget. Első látásra ez nem különbözik egy fényvédő réteggel ellátott szemüvegtől. Azonban ez többet tud! Különleges lencséje kiszűri a káros sugarakat. A kisszövetkezet munkatársai tudják, hogy a szemüveget egyébként nem viselők ellenérzéssel fogadják e hasznos kiegészítő eszközt, ezért tetszetős és kényelmes keretbe helyezték a szemet óvó speciális lencséket.

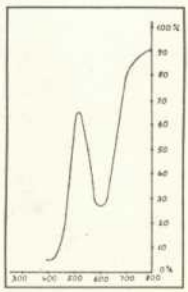
A dioptriás szemüveget viselők sincsenek kizárva, mert részükre a bevonat szintén elkészíthető.

Én magam is sokat dolgozom számítógéppel, illetve ő is nagyon sokat nyúz engem. Kipróbáltam ezt az új magyar szabadalmat, és mindenkinek ajánlom, aki képernyő előtt ül. A számítástechnikához a gépek, szoftverek és perifériák mellett hozzátartoznak az egészséget védő kiegészítő eszközök. Gondoljunk a magunk és gyermekeink épségére!

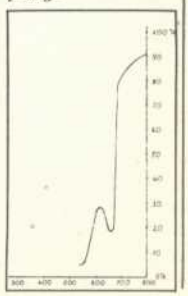
1. ábra. A CompuDrug szintelen, UV-szűrő lencséjének fényáteresztő képessége



2. ábra. A CompuDrug zöld, UV-szűrő lencséjének fényáteresztő képessége



3. ábra. A CompuDrug barna, UV-szűrő lencséjének fényáteresztő képessége



COMMODORE 16

Korszerűbb!...

A Magazin 1986. júliusi számában Pál László cikkében láttam egy APPEND programot. Azóta foglalkoztat a gondolat, hogy C16-ra olyan APPEND-et írjak, amely mindent elvégez, a felhasználónak menet közben nem kell még ezt-azt csinálnia.

Az APPEND-del programokat lehet összefűzni, nagyon egyszerűen. Töltsük be az APPEND-et, majd indítsuk el. Ha nincs a DATA sorokban hiba, akkor kitörli a BASIC-betöltőt, ezért akik először írják a programot, futtatás előtt mentsék el. Ezután az összefűzni kívánt programok közül az elsőt töltsük be, majd nyomjuk meg az F1 billentyűt. Indítsuk el a magnót. Betöltődik a következő program. Az APPEND összefűzi és 10-től tízesével átsorszámozza a programokat.

Hasznos lehet, ha szubrutinonként vannak megírva programjaink, és ezeket akarjuk egy egésszé összeállítani.

Arra vigyázzunk, hogy grafikus képernyőt és RS232 puffert ne használjunk; ezen a területen helyezkedik el a gépi kódú rész.

Ha az F1 billentyűt átprogramozzuk, az APPEND a SYS685 paranccsal hívható.

VARGA SÁNDOR

```
10 REM************************
20 REM*                      *
30 REM*        APPEND        *
40 REM*                      *
50 REM*    VARGA SANDOR      *
60 REM*                      *
70 REM************************
80 FORI=685TOI+21:READA:B=B+A:POKEI,A:NEXT
90 FORI=1015TOI+55:READA:B=B+A:POKEI,A:NEXT
100 IFB<>7777THENPRINT"ADAT HIBA":END
110 KEY1,"SYS685"+CHR$(13)
120 NEW
130 DATA 56,165,45,233,2,133,43,176,10
140 DATA 165,46,56,233,1,133,44,76,251,3
150 DATA 76,247,3
160 DATA 165,46,133,44,169,0,32,189
170 DATA 255,166,43,164,44,169,0,32,213
180 DATA 255,169,1,162,16,133,43,134,44
190 DATA 169,82,162,69,160,78,141,39,5
200 DATA 142,40,5,140,41,5
210 DATA 169,117,162,13,141,42,5,142,43
220 DATA 5,169,5,133,239,96
```

ZX-SPECTRUM

Beta BASIC

A program a Beta BASIC 3.1-es változatában készült, és mint önálló eljárás, tetszés szerint használható saját programjainkban. A rutin hívása például

1000 be n$, 10, 2, 10

formában történhet, amikor a 10-es sor 2. karakterétől kezdődően jelöli ki a rutin a beviendő füzér helyét, 10 karakter hosszan. Ha itt van a képernyőn valami, akkor a rutin ENTER hatására egyszerűen a kijelölt helyen szereplő karaktereket helyezi az n$ füzérbe, egyébként pedig a bebillentyűzött karaktereket. A 20-as sor állítja be azokat az értékeket, amelyeket a felsorolt változók felvesznek, ha például

1000 be j$

formában hívnánk a rutint.

Az eljárást olyan esetekben ajánlom, amikor sok „bebillentyűzendő" adatot kell feldolgozni: ilyenkor ugyanis a szerkesztősorból történő bevitelnél sokkal látványosabb és egyszerűbb mindjárt a megfelelő helyre írni a beviendő füzért. Ismétlődő adatokat nem kell újból bevinni, mert ENTER hatására az adott helyre előzőleg beírt adatok automatikusan „beolvasódnak".

Commodore-tulajdonosok! Örömmel látnék hasonló adottságú eljárást valamelyik Commodore gépre is.

RÁCZ MIKLÓS

```
10  DEF PROC be REF i$,x,y,l
    REM  RaMsOFT 1987
20  DEFAULT x=0,y=1,l=7,i$=""
30  PRINT AT x,y-1; FLASH 1;">";AT x,y+1; FLASH 1;"<";AT x,y;
40  GET z$
50  IF CODE z$=13 THEN
      FOR i=y TO y+l-1
        LET i$=i$+SCRN$(x,i)
      NEXT i
      GO TO 140
    ELSE LET i$=z$
60  PRINT STRING$(l," ");AT x,y;z$;
    BEEP .1,-20
70  DO
      PRINT "_";CHR$ 8;
      PAUSE 2
      LET z$=INKEY$
      PRINT " ";CHR$ 8;
      PAUSE 2
    LOOP  WHILE z$=""
80  IF z$=CHR$ 12 THEN
      IF LEN i$>0 THEN  BEEP .035,-10
        PRINT CHR$ 8;" ";CHR$ 8;
        LET i$=i$( TO LEN i$-1)
        GO TO 70
      ELSE  BEEP .035,10
90  IF z$=CHR$ 13 THEN  GO TO 140
100 IF z$<>CHR$ 12 THEN
      LET i$=i$+z$
      BEEP .035,-10
      PRINT z$;
110 IF LEN i$<l THEN  GO TO 70
    ELSE  GET z$
      BEEP .035,10
120 IF z$<>CHR$ 13 AND z$<>CHR$ 12 THEN  LET i$=i$( TO LEN i$-1)
      PRINT CHR$ 8;
130 GO TO 80
140 PRINT AT x,y-1;" ";AT x,y+1;" "
150 END PROC
```

Már nem bunda

A sportélet változására, korszerűsítésére is kihat hétköznapjaink technikai fejlődése. Az új helyzetben valamennyi sportágban elengedhetetlen a felkészítést és versenyzést segítő számítástechnikai rendszerek bevonása. A versenyek ésszerű életteervezéseket követelnek. Ezekhez azonban az eddig bevált, hagyományosan alkalmazott kiválasztási módszerek elavultak, így csak az egzakt adatokra támaszkodó vizsgálati eljárások vezethetnek eredményhez.

Ma már a fentiek ismertek, sőt mind szélesebb körben elfogadottak, egyre több területen alkalmazottak, noha a korszerű hazai sportinformatika és számítástechnika meglehetősen rövid múltra tekint vissza. Az úton elértünk már eredményeket, pedig csak a kezdetén vagyunk. Ahhoz, hogy a jelenleginél sikeresebbek legyünk, van még mit tennünk az elektronikus adatfeldolgozás, adattárolás, információtovábbítás terén. Nélkülözhetetlen — elsősorban IBM-kompatibilis gépekkel — a meglévő mikroszámítógépes állomány fejlesztése. És nem maradhat el annak a számítástechnikai központnak, országos adatbanknak a felállítása sem, amely csatlakozhatna a szocialista országok számítógépes és információs rendszeréhez, továbbá más adatbankokhoz is. A létrehozott adatbankok és információs számítóközpontok ugyanis jó szolgálatot tennének a sporttudományos kutatásoknak, a sportegészségügynek, a különféle nyilvántartásoknak és a sportéleti gazdálkodásnak.

Napjainkra nemcsak az ipar vagy a különböző tudományágak nem lehetnek meg számítógépes szoftverek nélkül, de a tömegsport és versenysport sem mellőzheti a számítógépek nyújtotta „szolgáltatásokat". Ahhoz, hogy nemzetközi mezőnyben szerepeljünk, ott eredményeket mutassunk fel, fontos valamennyi résztvevő módszerének elemzése, nyomon kísérése, és ha arra mód van, a sikerek adaptálása. Ugyanis a számítástechnikát kiválóan alkalmazzák külföldön a sportban, mint jól hasznosítható eszközt és módszert. Az ötletet az élet produkálta, nevezetesen, hogy a számítógépek sok területen szolgálják, segítik az embert — miért ne tennék ugyanezt a sportban?

Hogy mennyire bevált, arra nézzünk néhány példát, esetlegesen kiválasztott sportágat; olyat, ahol már sikerrel teremtettek kapcsolatot a számítógéppel.

A férfitornában használatosak már mikroszámítógépekhez írt programok. Ezek közül a kettős tagozódású logikai rendszer személyi állományt kezel, dolgoz fel, a tornaversenyek információs és pontérték-ellenőrző feladatait végzi, illetve a pontozóbírók munkáját ellenőrzi, értékeli és dokumentálja. Az ötven bíróra és kétszáz versenyzőre elkészített program nemcsak a torna világára alkalmazható, hanem némi változtatással más, pontozásos sportágban is hasznosítható. Segítségével ugyanis kiszűrhető a szubjektivitás és összehasonlítható a zsűritagok munkája. Sőt — ami nagyon fontos —, lehetővé teszi a legjobb bírók kiválasztását!

Az edzések megtervezése a majdani eredmények alfája. Mindehhez a mikroszámítógép partner, elengedhetetlen társ. E kijelentés igazolásául annyit, hogy manapság rendkívül nehéz az edzések optimális mennyiségét, intenzitását meghatározni. De a sportmozgások rögzítésével, számítógépes regisztrálásával, az adott sportági technika kialakításával előbbre léphet a szakember. Hogyan? A számítógépes mozgáselemzéssel pontos, részletes információhoz juthat és így az adatok birtokában kiküszöbölheti az esetleges hibákat.

Az úszás kedvelt, tömegeket vonzó sportág. Szerelmesei közül jó néhányan világklasszisok, csúcseredményt döntők lettek, öregbítve a magyarok hírnevét a nemzetközi sportéletben. Viszont cseppet sem mellékes az, hogy a sikeres versenyzés mellett hogyan, milyen színvonalon rendezünk meg egy-egy versenyt.

A mikroszámítógépekre írt programok ez utóbbit segíthetik. Elvégzik a különböző feladatokat, a kötelező adminisztrációt, a futam- és pályabeosztást, elkészítik a helyezési pontszámok alapján az egyesületi szinten részletezett, táblázatos pontszámösszesítést. A tömegsportban, mint a Balaton-átúszás, hagyományos versenyen az adatnyilvántartáson kívül eredményt mér és közöl, továbbá statisztikát értékel.

Maradva a víznél, evezzünk át egy másik, közkedvelt sportágba, ahol a gyakorlatban jól bevált a kajakos mozdulatát mérő program. Ebben terhelés alatt folyamatosan rögzítik a húzás erősségét, sebességét, a hajónak átadott erőt. A mikroszámítógép segítségével meghatározzák csapásonként az evező húzásisebesség-maximumát, hosszát, a lapáterő maximumát és impulzusát, teljesítménymaximumát, a hajónak átadott impulzus- és erőmaximumot. A nyert adatokat a program tetszőleges időintervallumra összegzi, és meghatározza a csapásszámot. A művelet végén grafikusan megjeleníti a kapott értékeket.

A mikroszámítógép a különböző labdajátékoknál is bevált. Programokkal megkönnyítik többek között a taktikázás tanítását úgy, hogy a sportolónak különféle taktikai variációkat mutatnak be. A reakcióidőt és figyelemkoncentrációt szintén programban dolgozzák fel, és az így nyert ismeretek birtokában objektíven mérik a rajtállapotban előálló pszichés összetevőket. E megfi-

Bunda...

gyelések jó szolgálatokat tesznek a versenyzők felkészítésében.

Hosszasan lehetne sorolni, hogy a legkülönbözőbb sportágakban miként hasznosítható a mikroszámítógép. Mi azonban a fentiekkel csupán ízelítőt kívántunk nyújtani a gyors, pontos sportinformatika eszközéről.

Beszámolónkat mégsem fejezhetjük be anélkül, hogy néhány szóval kitérnénk arra, hogy a bírók munkáját, kiválasztását miként ellenőrzi, könnyíti meg a részükre készített program. Hogy mennyire nem mellékes a versenyzés során, az eredménynél a bíró személye, egyénisége, felkészültsége, azt a laikusok is, mint drukkerek, jól tudják. Mert hiába a sportoló felkészültsége, tudása, az elfogult bíró sokszor juttathatja a jót is hátrányos helyzetbe. De egy mikroszámítógépre írt rendszerrel sikerült például a kézilabda-mérkőzésekre úgy kiválasztani a játékvezetőket, hogy a személyük elleni reklamációk száma egy év alatt hatodára csökkent. Korábban sok volt a panasz, a vita a bírók ítélete miatt. A számítógép közreműködését már nem kérdőjelezték meg a játékosok; belátták, hogy vele kizárták a szervezők a szubjektivitást, hiszen a képernyőn megjelenő adatok objektíveknek tekinthetők.

Az élet bebizonyította, hogy létjogosultságot nyert a sportban is a számítógép. Éppen ezért a jövőben ki kell szélesíteni a számítógép alkalmazását a sportélet valamennyi területén, mert már a kezdeti eredmények igazolták: a gép pontos, gyors és mentes mindenféle „bundától".

k. é.

1987. EVI TAVASZI FORDULOK

5 . HET NB1 FERFI 1

HODIKOT SE	-UJVAROSI KOHASZ	4	5	11	0	HALMAI JANOS	KOVACS ISTVAN
SZEGEDI VOLAN	-TATABANYAI BANY	4	6	18	0	HUMMEL PAL	KISS LASZLO
FERENCVAROSI TC	-DELEP SC	4	5	16	0	KAPONYI JANOS	KIS KAROLY
VARPALOTAI BANY	-RABA ETO	4	5	15	30	KEMECSEI ISTVAN	KOUTNY LAJOS
ELEKTROMOS SE	-BP.HONVED SE	4	5	11	0	KISS II ISTVAN	SZABO LASZLO
V.A.E.V. SC	-PECSI MSC	4	6	11	45	MATRAI ATTILA	SCHALLI ADAM
BCSABAI ELORE	-DEBRECENI DOZSA	4	5	11	0	KLIMENT JANOS	PAPP GYORGY

6 . HET NB1 FERFI 1

BP.HONVED SE	-V.A.E.V. SC	3	30	11	0	BERGMANN NANDOR	SZIKLAI ANDRAS
RABA ETO	-ELEKTROMOS SE	3	25	17	30	KAPONYI JANOS	KIS KAROLY
DELEP SC	-VARPALOTAI BANY	3	29	11	0	KEMECSEI ISTVAN	KOUTNY LAJOS
TATABANYAI BANY	-FERENCVAROSI TC	3	28	17	45	HUMMEL PAL	KISS LASZLO
UJVAROSI KOHASZ	-SZEGEDI VOLAN	3	28	17	15	AMBRUS ZOLTAN	HUCKER FERENC
DEBRECENI DOZSA	-HODIKOT SE	3	28	17	0	HAIDE RUDOLF	TEKAUER PETER
BCSABAI ELORE	-PECSI MSC	3	29	11	0	BRECSKA MIHALY	SZABO JOZSEF

7 . HET NB1 FERFI 1

HODIKOT SE	-UJVAROSI KOHASZ	4	5	11	0	KAPONYI JANOS	KIS KAROLY
SZEGEDI VOLAN	-TATABANYAI BANY	4	6	18	0	BERGMANN NANDOR	SZIKLAI ANDRAS
FERENCVAROSI TC	-DELEP SC	4	5	16	0	ANDORKA SANDOR	SCHOBER OTTO
VARPALOTAI BANY	-RABA ETO	4	5	15	30	GOZSY GABOR	VOLGYI LASZLO
ELEKTROMOS SE	-BP.HONVED SE	4	5	11	0	AMBRUS ZOLTAN	HUCKER FERENC
V.A.E.V. SC	-PECSI MSC	4	6	11	45	HUMMEL PAL	KISS LASZLO
BCSABAI ELORE	-DEBRECENI DOZSA	4	5	11	0	HALMAI JANOS	KOVACS ISTVAN

8 . HET NB1 FERFI 1

V.A.E.V. SC	-BCSABAI ELORE	4	11	17	30	GYULAI GEZA	KESZTHELYI LASZ
PECSI MSC	-ELEKTROMOS SE	4	13	11	0	.....	.....
BP.HONVED SE	-VARPALOTAI BANY	4	13	11	0	HUMMEL PAL	KISS LASZLO

Integrált szoftver

A sorozat eddigi részeihez hasonlóan ismertetem az átíráshoz szükséges tennivalókat, és közlöm a következő program listáját. Ismét felhívom a figyelmet arra, hogy a többször ismétlődő átírnivalókat csak az első előfordulásuknál említem meg, ezért ha valamit nem találnak itt, azt a sorozat korábbi cikkeiben keressék!

Amennyiben valamelyik géptípusra kellő számú jelentkező akad, vállaljuk az átírást is. Az így elkészített programcsomagot — a klub hagyományainak megfelelően — igen mérsékelt költségtérítés ellenében terjesztenénk.

Az átírás

A 10-es sorban a törlések a tárban helyet foglalnak a sztringeknek. A legtöbb gépnél erre nincs szükség.

A 20-as sorban üres helyek, egyenlőségjelek és a 124-nek megfelelő ASCII kódok beadása történik egy-egy sztring megfelelő helyére.

A SOUND utasítás mindenütt elhagyható vagy más hanghatást kiváltó utasítással helyettesíthető.

Az 530-as sorban az EOF(−1) az adatok végét jelzi a szalagon. Ez a jel gépenként más és más!

A felső vesszőt ennél a gépnél a REM utasítás rövidítése (például az 1200-as, 1300-as sorban).

DR. SIMONYI ENDRE

```
10 GOTO9000
21 CLS2:PRINTé195.'          OESSZEFEESUELOE
*;:M=MEM
23 FOR X=1 TO 2000:NEXT X:I¤=INKEY¤
30 IF MEM>32000 THEN CLEAR30000:T=200 EL
SE CLEAR15000:T=100
70 CH¤="ö":PP=1
80 DIMT¤(T+2):OE=1:LM=10:LN=60:PL=56:FL=
6:PP=1:SS¤=" "+STRING¤(30,61):LP=66:PRIN
Té453,"A FORMAAT VAALTOZTATJA (I/N)
";:GOSUB680:IF A¤<>"N"THEN IP=1:PRINTé48
5,"OLDAL SZUENET (I/N) ";:GOSUB680:IF A¤
<>"I" THEN PP=0
90 PRINTé485,STRING¤(21,32);:PRINTé453,"
SOROS, VAGY PAARHUZAMOS";:PRINTé485," <S
> OR <P>";:GOSUB680:IF A¤="S"THEN POKE&H
3FF,1 ELSE POKE&H3FF,0
100 CLS:GOSUB 6000:CLS:PRINT:PRINTTAB(2)
"CIMADATBAZIS-TOELTEES":PRINTSS¤:PRINT"A
Z ADATBAZIS EDDIGI NEVE: POSTAAZO":PRINT
:LINE INPUT"UJ NEEV>? ";AF¤:IF AF¤="POST
AAZO"
110 PRINT:PRINTAF¤;"- KERESEESE":PRINT:G
OSUB500:GOTO 1000
500 OPEN"I",-1,AF¤:IF EOF(-1) THEN PRINT
"AZ ADAT NINCS AZ ADATBAZISBAN":END ELSE
 RETURN
680 A¤=INKEY¤:IF A¤="" THEN 680 ELSE RET
URN
1000 CLS:IF EOF(-1) THEN END
1010 LINE INPUT#-1,R¤
1020 GOSUB 1600:AF=AF+1:CLS:PRINT:PRINT"
NYOMTATAASI SZAAM";AF:PRINTSS¤:PRINT"A M
AGNOT NEM SZABAD KIKAPCSOLNI!";:GOTO 700
0
1600 GOSUB1710:LN¤=LEFT¤(R¤,P-1):GOSUB17
00:GN¤=LEFT¤(R¤,P-1):GOSUB1700:TT¤=LEFT¤
(R¤,P-1):GOSUB1700:A1¤=LEFT¤(R¤,P-1):GOS
UB1700:A2¤=LEFT¤(R¤,P-1):GOSUB1700:CS¤=L
EFT¤(R¤,P-1):GOSUB1700:PH¤=LEFT¤(R¤,P-1)
:GOSUB1700:CD¤=R¤:R¤=""
1650 L1¤=TT¤+" "+GN¤+" "+LN¤:L6¤=TT¤+" "
+LN¤:L2¤=A1¤:L3¤=A2¤:L4¤=CS¤:RETURN
1700 R¤=RIGHT¤(R¤,LEN(R¤)-P)
1710 P=INSTR(1,R¤,CH¤):RETURN
2000 IF L1¤<>""THEN PRINT#-2,TAB(LM)L1¤:
CL=CL+1
2010 IF L2¤<>""THENPRINT#-2,TAB(LM)L2¤:C
L=CL+1
2020 IF L3¤<>"" THEN PRINT#-2,TAB(LM)L3¤
:CL=CL+1
2030 IF L4¤<>"" THEN PRINT#-2,TAB(LM)L4¤
:CL=CL+1
2040 IF RIGHT¤(A¤,1)="e" AND L6¤<>" " TH
EN GOSUB 2160:PRINT#-2," ":PRINT#-2," ":
PRINT#-2,TAB(LM)"Tisztelt";L6¤:CL=CL+3
2100 GOTO 7080
2160 P=1:P=INSTR(P,L6¤,","):IF P<>0 THEN
 L6¤=LEFT¤(L6¤,P):RETURN
2170 P=LEN(L6¤):FOR WW=P TO 1 STEP-1:IF
MID¤(L6¤,WW,1)<>" " THEN P=WW:WW=1:GOTO2
180
2180 NEXT WW:L6¤=LEFT¤(L6¤,P)+",":RETURN
6000 CLS:PRINT:PRINT"SZOEVEGADATBAZIS-TO
ELTEES":POKE329,255:PRINTSS¤
6020 INPUT"ADATBAZIS NEVE>";DS¤:DS=0:LI=
0:IFDS¤="U"THEN END
6040 FORX=0 TO T:T¤(X)="":NEXTX:OPEN"I",
-1,DS¤:INPUT#-1,LM,LN,PL,FL,PG,OE,HD¤,PI
,JY
6070 FOR X=0TO T:LINEINPUT#-1,T¤(X):IF E
OF(-1)THEN X=T
6100 NEXTX
6110 CLOSE-1:RETURN
7000 P1=PG:PS=0:SI=0:TAB=LM:N=0
7050 OPEN"O",-2,"PRINTER":IF AL¤="BE"THE
N AL¤="":GOTO7080ELSEGOSUB7720
7080 A¤=T¤(N)
7090 IF LEFT¤(A¤,1)=" "THENA¤=RIGHT¤(A¤,
LEN(A¤)-1):GOTO7090
7100 N=N+1:IFN > T THEN7730ELSE IF CL>=
PL ANDLEFT¤(A¤,2)<>"/f"THEN A¤="  "+A¤:G
OSUB7670
7120 IF LEFT¤(A¤,1)<>"/"THEN 7430ELSE IF
 LEFT¤(A¤,2)="/d"THEN 7080
7130 IFLEFT¤(A¤,2)="/k" THENPRINTA¤:LINE
INPUTA¤:GOTO7120
7140 IFLEFT¤(A¤,2)<>"/a"THEN7220ELSE PC¤
=MID¤(A¤,3,3):PC=VAL(PC¤)
7160 IFPC>255THENPRINTé228,"SEGEEDKOOD":
GOTO7080
7170 PRINT#-2,CHR¤(PC);:PC¤=STR¤(PC):X=L
```

```
EN(PC¤)-1:A¤=RIGHT¤(A¤,LEN(A¤)-X-2):IF A
¤<>"THEN7140
7210 PRINT#-2," ":CL=CL+1:GOTO7080
7220 IF LEFT¤(A¤,2)="/f"THENGOSUB7670:GO
TO7120
7230 IFLEFT¤(A¤,2)="/c"THEN GOTO 7810
7240 IFLEFT¤(A¤,2)<>"/i"THEN7290
7250 IF SI=1 THEN7270
7260 SI=1:LM=LM+5:LN=LN-10:GOTO7280
7270 SI=0:LM=LM-5:LN=LN+10
7280 A¤=RIGHT¤(A¤,LEN(A¤)-2):TAB=LM
7290 IF LEFT¤(A¤,2)<>"/p"THEN7320
7300 PRINT#-2," ":A¤=RIGHT¤(A¤,LEN(A¤)-2
):PS=1:TAB=TAB+PI:LN=LN-PI:CL=CL+1:IF DS
=1THENPRINT#-2," ":CL=CL+1
7310 GOTO7120
7320 IF LEFT¤(A¤,2)<>"/s"THEN7360
7330 X¤=MID¤(A¤,3,2):X=VAL(X¤):FOR Y=1TO
 X:PRINT#-2," ":CL= CL+1:NEXTY:X¤=STR¤(X
):X=LEN(X¤)-1:A¤=RIGHT¤(A¤,LEN(A¤)-X-2):
GOTO7120
7360 IF LEFT¤(A¤,2)<>"/t"THEN7410
7370 X¤=MID¤(A¤,3,2):TAB=VAL(X¤):X¤=STR¤
(TAB):X=LEN(X¤)-1:IF TAB<0 THEN X=X+1
7390 A¤=RIGHT¤(A¤,LEN(A¤)-X-2):TAB=TAB+L
M:T¤=A¤:L=LEN(A¤):GOTO7430
7410 IF LEFT¤(A¤,2)<>"/l"THEN7420ELSE A¤
=RIGHT¤(A¤,LEN(A¤)-2)
7420 IF LEFT¤(A¤,2)="/m" THEN 2000
7430 IF LEN(A¤)>LN THEN7480
7440 ZZ¤=A¤:GOSUB8200:A¤=ZZ¤:IF T¤(N)=""
THEN PRINT#-2,TAB(TAB)A¤:GOTO7780
7450 IF LEFT¤(T¤(N),1)="/"THEN T¤=A¤:L=L
EN(A¤):GOTO7550
7460 IF LEN(A¤)+LEN(T¤(N))>250THENGOSUB8
040
7470 A¤=A¤+T¤(N):N=N+1:IFN > T THEN7780E
LSE7430
7480 IF LEFT¤(A¤,1)=" "THEN A¤=RIGHT¤(A¤
,LEN(A¤)-1):GOTO7480
7490 L=1:LL=L
7500 LL=INSTR(LL,A¤," "):IF LL= 0AND LEN
(A¤)<=LN THEN L=LEN(A¤):GOTO7540
7520 IF LL= 0AND LEN(A¤)> LN THEN7540
7530 IF LL <= LN+1 THEN L = LL:LL=LL+1:G
OTO7500
7540 T¤=LEFT¤(A¤,L)
7550 IF LEFT¤(T¤,1)=" "THEN T¤=RIGHT¤(T¤
,LEN(T¤)-1):GOTO7550
7560 IF CL>= PL THEN A¤="  "+A¤:GOSUB767
0
7570 IF LI =1 THENPRINT#-2,N-1;
7580 IF JY=1 THENGOSUB8110
7590 ZZ¤=T¤:GOSUB8200:T¤=ZZ¤:PRINT#-2,TA
B(TAB);T¤:CL= CL+1:IF DS=1THENPRINT#-2,"
 ":CL=CL+1
7600 PRINTé167,"SOROK SZAAMA";CL;"KINYOM
TATVA":PRINTé288,STRING¤(99,32):PRINTé25
7,T¤:TAB = LM:IFPS=1 THENLN=LN+PI:PS=0
7610 QT¤=INKEY¤:IF QT¤="U"OR QT¤="u"THEN
7780
7630 A¤=RIGHT¤(A¤,LEN(A¤)-L):IF LEN(A¤)>
 LN THEN 7480
7650 IF A¤=""THEN7080
7660 GOTO7440
7670 IF IP=1THENPRINT#-2,CHR¤(12):GOTO7
690
7680 FORF = CL+FL TOLP:PRINT#-2," ":NEXT
F
7690 A¤=RIGHT¤(A¤,LEN(A¤)-2)
7700 IF PP=1 THEN SOUND120,18:PRINTé228,
"A FOLYTATAASHOZ LENYOMANDO AZ <ENTER>-G
OMB";:GOSUB680:PRINTé228," "
7720 IFFL>1THENFOR X=1TO FL:PRINT#-2," "
:NEXTX
7730 CL=1:IF PG=0 THEN RETURN
7740 IF PG=1THEN PG=PG+1:RETURN
7745 IF SI=1THEN LM=LM-5:LN=LN+5
7750 ZZ¤=HD¤:GOSUB8200:HD¤=ZZ¤:PRINT#-2,
TAB(LM)HD¤;:PRINT#-2,TAB((TAB+LN)-7)"Old
al";PG:CL= CL+2:IF DS=1 THEN PRINT#-2,"
":CL=CL+1
7755 IF SI=1THEN LM=LM+5:LN=LN-5
7770 PG=PG+1:PRINT#-2," ":RETURN
7780 IF OE=1AND IP=1THENPRINT#-2,CHR¤(12
);
7785 IF OE=1ANDIP=0THENFORX=CL+FL TO LP-
1:PRINT#-2," ":NEXTX
7790 IF PS=1 THEN LN=LN+PI:PS=0
7800 CLOSE-2:PG=P1:X=0:N=0:GOTO1000
7810 A¤=RIGHT¤(A¤,LEN(A¤)-2):ZZ¤=A¤:GOSU
B8200:A¤=ZZ¤:PRINT#-2,TAB((LN-LEN(A¤))/2
+LM)A¤:CL=CL+1:IF DS=1THENPRINT#-2," ":C
L=CL+1
7820 GOTO7080
8040 PRINTé228,"KIVAAGANDO A HOZZAARENDE
LEESIG";N:FOR X = T TO N+1STEP-1:T¤(X)=T
¤(X-1):NEXTX:X=LEN(T¤(N))/2:X=INT(X):T¤(
N+1)=RIGHT¤(T¤(N),LEN(T¤(N))-X):T¤(N)=LE
FT¤(T¤(N),X):RETURN
8110 IF T¤=A¤ AND LEN(A¤) < LN-7 OR LEN(
T¤) < LN-12 THEN RETURN
8120 IF RIGHT¤(T¤,1)=" "THEN T¤=LEFT¤(T¤
,LEN(T¤)-1):GOTO8120
8130 P=RND(LN)
8140 P=INSTR(P,T¤," "):IF LEN(T¤)>=LN TH
ENRETURN
8160 IF P=0 THEN P=1:GOTO 8140
8170 T¤=LEFT¤(T¤,P)+" "+RIGHT¤(T¤,LEN(T¤
)-P):P=P+3:GOTO8140
8200 ZY¤="":FORX=1TO LEN(ZZ¤):ZX¤=MID¤(Z
Z¤,X,1):IF (ZX¤<>"%")AND(ZX¤<>"¤")AND(ZX
¤<>"&")AND(ZX¤<>"#") THEN8250
8210 IF ZX¤="#" THEN ZX¤=CHR¤(125):GOTO8
250
8220 IF ZX¤="%" THEN ZX¤=CHR¤(124):GOTO8
250
8230 IF ZX¤="¤" THEN ZX¤=CHR¤(96):GOTO82
50
8240 ZX¤=CHR¤(126)
8250 ZY¤=ZY¤+ZX¤:NEXTX:ZZ¤=ZY¤:RETURN
9000 CLS0:PCLEAR1:GOTO21
9999 MOTORON:FOR X=1 TO 8000:NEXT X:FORX
=1 TO2:CSAVE"FEESUELO":MOTORON:FOR Z=1 T
O 999:NEXT Z:NEXT X:MOTOROFF
```

Primo

Két botkormánnyal

Váradi Gábor (Kiskörös) írja: „A Primóba beépítettem a szükséges A, B csatlakozót, a hozzájuk kapcsolódó áramkörökkel együtt. Kérem, alakítsák át, hogy alkalmas legyen a géphez. (Azért kettő, mert két srácom van, és időnként együtt akarnak játszani.)

Kérem, jelölje meg, hogy melyik a jobb, és a bal, majd készítek nekik egy kis elosztódobot a géphez, hogy együtt és külön is tudjuk használni."

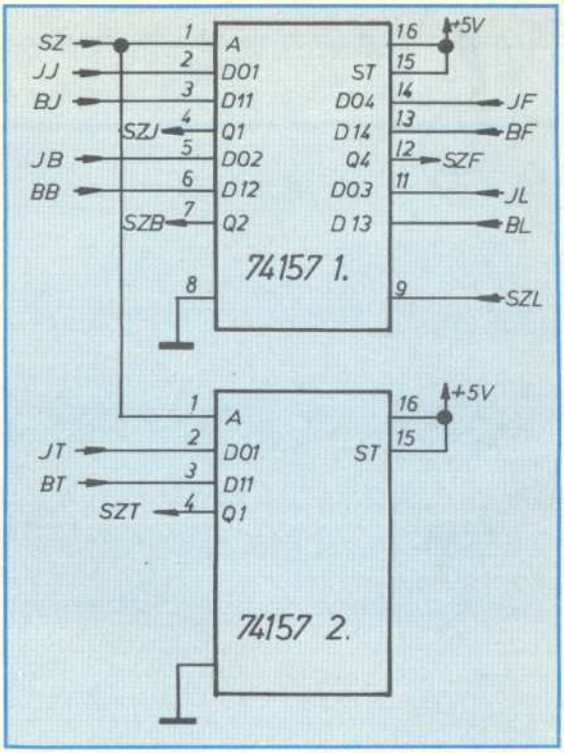
Az általam közölt cikk (és a benne levő program) szerint mindkét csatlakozót használja. Így nincs jobb és bal, hanem csak egyetlen botkormány. A két botkormányos esetben a programnak kell váltogatni, vagy úgy, hogy:

— mindig mindkét botkormány párhuzamosan van kötve, és a kezelők váltogatva kapcsolnak (ez nagyon egyszerű, de nem célszerű),

— vagy úgy, hogy egy két integrált áramkörből álló „kapcsolást" építünk.

Ez utóbbit ismertetem. A bemeneteknél a J, B mint első betű a botkormányt jelöli, a J, B, F, L, T mint második betű az irányokat és a tűzgombot jelöli. Az A bemenet a kiválasztó, az ST az engedélyező. Az SZ jelzi a számítógépet. Mivel a második alkatrészen használatlan három kapcsolási lehetőség, ezért további funkciók (például folyamatos tüzelés) is megvalósítható.

— SE —



ADOK—VESZEK—CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közületeknek gépelt soronként (60 karakter) 100,- Ft, magánszemélyeknek az első sor 50,- Ft, minden további sor 20,-Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

ADOK

Angol nyelvvizsga-előkészítő szótanító program C64-re. 11 témakörben 1400 szót tanít, kiejtéssel. 2-3-szor gyorsabban tanulhat, mint szótárból. Ára magánszemélyeknek és közületeknek floppyn 3600 forint. Kívánságra részletes tájékoztatót küldök. Tóth András számítástechnika, Post Restant, 1364 Budapest 4.

C64-hez Geos program magyar fordítása eladó. Vidos Nagy Leona, Budapest, György Aladár u. 30. 1125

C64-hez bővítő portra csatlakoztatható reset+super reset gomb 220 forintért eladó. Utánvéttel elküldöm. Kérjen tájékoztatót! Tárkány László, Szeged, Rózsa Ferenc sgt. 128/a. 6728

Commodore 64 programok titkosítását floppyn vállalom magánszemélyeknek és közületeknek. Minden másolóval szemben biztos védelem, igen nehéz visszafejthetőség. Levélben részletesen tájékoztatom. Báldi András, Post Restant, 1364 Budapest 4.

C64 és C128-tulajdonosok, figyelem! Junoszty televíziót videomonitorrá alakítok (kép+hang), az eredeti funkciók megtartása mellett. C128 esetén 80 oszlopos megjelenítésre is alkalmas. Páll Miklós, Budapest, II. Rákóczi Ferenc u. 34/5. K.7-10 C lépcső. 1214

SHARP PC-1247, Sharp CE-125 thermoprinter/microcasette recorder/outer cassette interface (ez utóbbi PC-1245-51-hez is jó) külön is eladó. Dr. Nagy András, Szeged, Aradi vértanúk tere 3. 6720

80 k-s Spectrum beépített hangmodulátorral, intelligens botkormány-interfésszel, 400 programmal eladó. Czulák Tamás, Pécs, Siklósi út 28. 7632. Telefon: 72/41-114

Videoton TVC tartozékokkal, memóriabővítővel, programokkal eladó. Irányár 10 ezer forint. Bíró János, Fót, Kinizsi u. 12. 2151

48 k-s Spectrum, tanítható botkormány interfész, botkormány, 150 darab válogatott program, nagyon sok irodalom eladó. Az egész együtt 17 000 forint. Vantai Dezső, Sajószentpéter, Móra Ferenc u. 20.

48 k-s ZX-Spectrum+ sürgősen eladó. Tartozékok: joystick turbo interfész, 400 darab program és bő szakirodalom. Hangyási Zoltán, Újkígyós, Petőfi u. 53/1. 5661

Stop! Már megint a hirdetéseket böngészi? Nyilván szórakoztató, izgalmas játékprogramokat keres. Nálam megtalálja! Kérje ingyenes tájékoztatómat! Kurunczai József, Szatymaz, Pf. 1044. 6763

Másoló és hibajavító program. Teljes lemezmásolás, formálás, ellenőrzés, hibajavítás 6 perc alatt. Bővebb felvilágosítás: Pivarnyik Attila vagy Szakál László, Miskolc-Egyetemváros, Vörös hadsereg u. 110. Telefon: 46/65-111

Programok C Plus/4 és C16 számítógépekre: FORTH nyelv, turbo file kazettás fájlkezelő, disk system BASIC és assembler fejlesztő program, Copy 226 lemezes/szalagos másoló, C64/C Plus/4 turbo, C64/C Plus/4 rutin és rendszerváltozó összehasonlító táblázat, Profiass C Plus/4 kétmenetes assembly fordító. A programok ára 180-600 forint, kazettán, utánvéttel, részletes kezelési utasítással. Kérésre ismertetőt küldök. Pelsőczi Gyula, Szilasliget, Ady Endre u. 36. 2145

64 kbájtos ZX81 sok programmal, botkormánnyal eladó. Árajánlatot levélben kérek. Vigh György, Budapest, Margit u. 8. 1165

VESZEK

Sürgősen vennénk 1 vagy 2 darab CP/M 2.2\ operációs rendszer alatt működő Apple IIe számítógéphez csatlakoztatható 10 Mbájtos vagy ennél nagyobb kapacitású winchester-tárolót. Ajánlatokat a Geofizikai Kutató Vállalat beruházási osztályára, Andor Ernő névre kérünk, írásban vagy telefonon (Budapest, Népköztársaság útja 59. Telefon: 221-050).

CSERÉLEK

C16, C Plus/4 programokat cserélnék. Kapcsa Gyula, Debrecen, Kőfarago u. 21. 4033

C16 és C Plus/4 programokat cserélek. Farkas Tibor, Zalaszentgrót, Felszabadulás u. 1. 8790

C64-es programokat cserélek kazettán. Kecskés Zsolt, Debrecen, Barna u. 3. 4025

Commodore 128-as gépre és 1571-es floppyra felhasználói programok - adatfeldolgozó, szövegszerkesztő stb. - cseréjéhez kapcsolatot keresek. A programok kézikönyve, leírása is érdekel. Gömöri József, Budapest, Lujza u. 1/b. 1086

Primo programokat cserélek. Várom olyanok jelentkezését, akik hardverbővítést végeztek gépükön. Varsányi Gábor, Nagyatád, Árads u. TX/c. 7500

MSX rendszerű Sanyo MPC 64 gépre és C64-re szoftvert cserélnék. Tóth Zsuzsa, Budapest, Katona József u. 33/b. 1137

µINFORM

A tartalomleírások az alábbi folyóiratokban megjelent programlistákról készültek:

A folyóirat neve	Kódja
64'er Magazin	64er
Chip Magazin	chip
Commodore Horizons	coho
Commodore Microcomputers	comi
Compute!	cute
Computer Persönlich	pers
Happy Computer	happ
hc - Mein Home-Computer	hc
mc - Zeitschrift	mc
Run /USA/	run
Sinclair User	sinc
Your Sinclair	ysin

A tartalomleíró szóláncokat permutáltuk, a szóláncvariánsokat pedig alfabetikusan rendeztük.

A tartalomleírás egy szóláncból áll, majd a listában ezt követi a forrás megjelölése a folyóirat azonosítójával, a megjelenés dátumával és a cikk előkereséséhez a kezdő oldalszám és a terjedelem megadásával. A mellékelt lista értelmezéséhez még az alábbiakat kell tudni. A tartalomleírás szóláncában elsőként a téma átfogó megnevezése, utána a számítógéptípus(ok), ezt követően a szűkebben jelölt tartalom meghatározása szerepel, majd esetlegesen néhány, a közleményt minősítő adat (például : cikksorozat).

A forráshely karaktersorozatát nyíl vezeti be, melyet a / jelig a folyóiratok azonosítója, a két / jel között az évszám, folyóiratszám és kötőjellel a kezdő oldalszám követi, a végén pedig a közlemény teljes oldalterjedelme áll.

A folyóiratok a SZÁMALK szakkönyvtárában (Budapest XI., Szakasits Á. út 68. Nyitva: 8-tól fél 5-ig. Tel.: 853-111/251) is föllelhetők. A kiválasztott anyagról másolat rendelhető az alábbi formában:

SZÁMALK Szakkönyvtára
Budapest, 112. Pf.: 146. 1502

Megrendelem a Mikroszámítógép Magazin 1987/ sz. alapján a következő folyóirat-oldal-másolatokat:

Kód: ________ Példányszám: ________
Kód: ________ Példányszám: ________
Kód: ________ Példányszám: ________

A megrendeléshez csatolom az oldalankénti 8,- Ft-os szolgáltatási díj befizetését igazoló csekkszelvényt.
Dátum, név, pontos cím.

```
PROGRAMLISTA
    assembler programozas▮cikksorozat▮c
    ommodore 64▮jatekprogram keszites
    -/happ/86.09-52/4
PROGRAMLISTA
    assembler programozas▮commodore 128
    ▮gepikod basic data-sorra alakitasa
    -/hc/86.09-82/1
PROGRAMLISTA
    atari xl▮'easy data'▮data-sor begepe
    lesi segedlet -/hc/86.09-45/2
PROGRAMLISTA
    atari xl▮listiator a rem-sorokban▮to
    mörites -/hc/86.09-46/2
PROGRAMLISTA
    atari xl/xe▮grafika▮matematika▮forga
    stestek▮turbo-basic
    -/happ/86.09-78/2
PROGRAMLISTA
    atari xl/xe▮hosszu basic programok
    betoltese gepi kodu segitseggel
    -/hc/86.09-49/2
PROGRAMLISTA
    atari xl/xe▮katalogusbetoltes
    -/hc/86.09-47/3
PROGRAMLISTA
    atari xl/xe▮programozasi trukkok▮(tu
    rbo-windows) -/happ/86.09-33/1
PROGRAMLISTA
    atari▮data-sor begepelesi segedlet
    -/hc/86.09-50/1
PROGRAMLISTA
    basic programozas▮commodore 64▮(dump
    )▮tombok es rendszer-valtozok kiirata
    sa -/64er/86.09-75/1
PROGRAMLISTA
    basic programozas▮commodore 64▮grafi
    ka▮utasitaskeszlet bovites
    -/run/86.09-58/9
PROGRAMLISTA
    basic programozas▮commodore 64/128▮a
    utomatikus data-sor generalas hexkod
    okbol▮gepi rutinok beepitese
    -/run/86.09-82/3
PROGRAMLISTA
    commodore 128▮fizika▮kemia▮oktatas▮r
    adioaktiv bomlas -/happ/86.09-56/4
PROGRAMLISTA
    commodore 128▮grafika▮matematika▮fug
    geny abrazolas -/run2/86.09-65/7
PROGRAMLISTA
    commodore 128▮programbetoltes katalo
    gusrol -/run2/86.09-47/2
PROGRAMLISTA
    commodore 128▮programdokumentalas▮cr
    ef-lista a 1541/70/71 floppykhoz
    -/64er/86.09-71/3
PROGRAMLISTA
    commodore 16▮analog-ora generalasa
    -/run2/86.09-50/2
PROGRAMLISTA
    commodore 16▮grafika▮polarkoordinata
    k hasznalata▮a circle sokoldalu felh
    asznalasa -/run2/86.09-105/5
PROGRAMLISTA
    commodore 64▮grafika▮hi-eddi▮nyomtat
    o rutinok -/64er/86.09-72/1
PROGRAMLISTA
    commodore 64▮hypra basic▮bovito modu
    lok▮renumber utasitas
    -/64er/86.09-76/3
PROGRAMLISTA
    commodore 64▮jatekprogram keszites▮b
    otkormany lekerdezes▮interrupt-kezel
    es -/hc/86.09-81/1
PROGRAMLISTA
    commodore 64▮jatekprogram keszites▮g
    orgetes-torles▮kijelolt kepernyoreze
    ktor kezeles -/hc/86.09-73/2
PROGRAMLISTA
    commodore 64▮jatekprogram keszites▮b
    illentyuzet lekerdezes gyorsitasa
    -/run2/86.09-44/3
PROGRAMLISTA
    commodore 64▮listazas letiltas▮run s
    zimulacio▮kodolt basic
    -/run2/86.09-41/4
PROGRAMLISTA
    commodore 64▮maszk szerkesztes▮sprit
    e formalas -/run2/86.09-34/6
PROGRAMLISTA
    commodore 64▮matematika▮egesz-szamos
    muveletek gyorsitasa
    -/run2/86.09-52/6
PROGRAMLISTA
    commodore 64▮muveletgyorsitas▮nyomta
    to(mps801) -/64er/86.09-34/2
PROGRAMLISTA
    commodore 64▮nyomtato(1520)▮hires gr
    afika nyomtatasa 64000pont 15perc se
    besseggel -/64er/86.09-33/1
PROGRAMLISTA
    commodore 64▮programbeolvasas byte-s
    orsz es abrazolas 4 fele kodban
    -/run2/86.09-81/3
PROGRAMLISTA
    commodore 64▮resettel torolt program
    ok visszanyerese▮gepi rutin
    -/happ/86.09-64/2
PROGRAMLISTA
    commodore 64▮szakaszos programlista-
    megjelenites -/chip/86.09-173/1
PROGRAMLISTA
    commodore 64/128▮jatekprogram▮(face
    of the bones) -/happ/86.09-60/5
PROGRAMLISTA
    commodore 64/128▮lemezegyseg(1570/71
    )▮muveletgyorsitas▮3-szoros sebesseg
    c64 modusban -/64er/86.09-50/5
PROGRAMLISTA
    commodore 64/vc20▮nagymeretu karakte
    rek nyomtatasa -/run/86.09-76/2
PROGRAMLISTA
    helyesbites▮(86.06)33/(86.07)30(30/1
    34)(86.08)82 -/64er/86.09-61/1
PROGRAMLISTA
    helyesbites▮(86.08)26/(86.07)32
    -/run2/86.09-40/1
PROGRAMLISTA
    robot▮vezerles▮commodore 64▮23 uj ba
    sic utasitas -/chip/86.09-143/5
PROGRAMLISTA
    sinclair spectrum▮(print at) gepi ru
    tin -/chip/86.09-147/2
PROGRAMLISTA
    sinclair spectrum▮ablakozas gorgetes
    ▮multitasking▮csak interface 1-gyel
    hasznalhato -/happ/86.09-50/6
PROGRAMLISTA
    sinclair spectrum▮grafika▮abrakeszit
    es 62 alapjelbol -/hc/86.09-76/2
PROGRAMLISTA
    sinclair spectrum▮jatekprogram▮kocka
    dobas szimulacio -/hc/86.09-72/2
PROGRAMLISTA
    sinclair spectrum▮karaktersorozat ge
    neralas -/ysin/86.09-61/1
PROGRAMLISTA
    sinclair spectrum▮lemezegyseg(discov
    ery)▮katalogus file hasznalata
    -/ysin/86.09-52/1
PROGRAMLISTA
    sinclair spectrum▮matematika▮grafiku
    s integralas -/hc/86.09-72/4
PROGRAMLISTA
    sinclair spectrum▮regiszter modosito
    utasitasok -/ysin/86.09-56/5
PROGRAMLISTA
    sinclair spectrum▮strukturalt progra
    mozas▮input rutin gepi kodban
    -/ysin/86.09-36/4
PROGRAMLISTA
    speed-dos▮commodore 64▮modositott je
    labrazolas▮szinillesztes
    -/hc/86.09-79/2
PROGRAMLISTA
    sprite▮commodore 64▮nagyitas/kicsin
    yites▮x- es y-tengely koruli forgatas
    -/64er/86.09-58/3
PROGRAMLISTA
    vonalkod▮commodore 64▮nyomtatas▮ean-
    kodok epson nyomtatoval
    -/64er/86.09-54/3
```

Az információ mennyiségét ugyanúgy ki lehet mérőszámmal fejezni, ahogy a tömeget, az időt stb. Az információmennyiség mértéke a *bit* (a binary digit — a bináris számjegy). Tudjuk: az információt a mennyiség önmagában még nem írja le teljesen, sőt nem is biztos, hogy maga a mennyiség információ értékű. Hiszen — bár a:

— Pista, magához megyek feleségül!

— Pista, elmegyek magával moziba.

két közlés, mely adott körülmények között egy-egy bit információmennyiséget hordozhat (másként fogalmazva: egy-egy bit bizonytalanságot szüntethet meg), mennyiségileg azonos, mégsem ezen múlnak a dolgok . . . Mert — amint Rényi is írja — mekkora a különbség a két közlés minőségében (tartalmában)!

Vannak, akik úgy vélik, az infomáció mennyisége és minősége valamilyen *bizonytalansági relációban* van egymással: a minőséget csak a mennyiség rovására lehet javítani. Amikor tehát az információt bitekkel mérjük, tudatában kell lennünk annak, hogy ezzel csak a jelenség egyik, nem is mindig a leglényegesebb oldalát ragadtuk meg.

A hír, közlés valamilyen helyzetre (az információ forrására) vonatkozó tudásunk *bizonytalanságát* csökkenti. Ezért Weaver nyomán helyes így fogalmazni: „Az információ egy üzenet kiválasztásában rejlő szabad választásunk mértékét jelenti . . . Félrevezető (bár gyakran kényelmes) azt mondanunk, hogy egyik vagy másik üzenet egységnyi információt hordoz. Az információ fogalma ugyanis nem egyedi üzenetekre vonatkozik (mint a jelentés fogalma), hanem sokkal inkább a helyzet egészére . . ."

Nézzünk egy példát. Tartózkodjon egy teremben 32 ember azzal a céllal, hogy elnököt választanak maguk közül. Induljon mindegyikük egyenlő eséllyel. A választás eredményhirdetése előtt még nem tudjuk, ki az elnök. Erre a helyzetre vonatkozó tudatlanságunk mérőszámát ilyen esetben az ún. Hartley-formulával számítjuk ki:

$$H = \log_2 N = \log_2 32 = 5 \text{ bit.}$$

(N a lehetséges egyenlő valószínűségű eredmények száma.)

Azt mondjuk, hogy a helyzetre vonatkozó tudatlanságunk (a bizonytalanság, másképp az elnök kiválasztására vonatkozó szabadság) mértéke H = 5 bit. Amikor közlik velünk az elnök nevét, ezt a bizonytalanságot szüntetik meg. Nem szabatos megfogalmazásban: az elnök személyére vonatkozó közlés információtartalma 5 bit. Miért éppen ennyi? Miért alkalmas a Hartley-formula ilyen helyzetek bizonytalanságának kifejezésére? Van-e valamilyen megfoghatóbb jelentése ennek az 5-ös számnak?

Van. A bizonyításra itt most nem térünk ki (az megtalálható az irodalomban), de elmondjuk, hogy az így kapott szám azonos azoknak a kérdéseknek a számával, amelyeket — a barkochba játék szabályai szerint — a leghatékonyabb stratégiának megfelelően fel kell tennünk ahhoz, hogy a bizonytalanságot teljesen eloszlassuk (a példánkban: kitaláljuk az elnök személyét). Próbáljuk ki, de a legcélszerűbb módon (minimális számú kérdéssel), hogyan tudjuk eltalálni az elnök személyét:

1. Az első 16-os csoportból választották meg? (A válaszból megtudjuk, melyik 16-os csoportból választották az elnököt. Ezt a tudást fejezi ki a következőkben az „előbbi" utalás, hiszen mindig úgy megyünk tovább, hogy az „előbbi" válaszból már konkrét ismeretünk van.)
2. Az „előbbin" belül az első 8-as csoportból választották? (A válasz után már tudjuk, hogy a 4 darab 8-as csoport közül melyikből választották az elnököt.)
3. Az „előbbin" belül az első 4-es csoportból választották?
4. Az „előbbin" belül az első 2-es csoportból választották?
5. A kettő közül az első lett az elnök?

Érdemes itt megjegyezni, hogy akkor is csak 5, jól megválasztott kérdés kell, ha a kérdéseket nem egymás után, hanem egyszerre szabad csak feltenni.

Ha két egyforma esélyű jelölt közül történik a választás, akkor már egyetlen (döntő) kérdésre adott válasszal teljesen meg lehet szüntetni a bizonytalanságot:

$$H = \log_2 2 = 1 \text{ bit.}$$

A Hartley-formula a célszerűen megválasztott kérdések számának meghatározására egyszerű receptet ad, melyet akkor alkalmazhatunk, ha az összes válasz (üzenet, állapot) *egyforma valószínűségű* (az összes jelölt esélye azonos). De mi van abban az esetben, ha H értéke nem egész szám, például a választás 3, egyenlő esélyű jelölt közül történik:

$$H = \log_2 3 = 1{,}58 \text{ bit.}$$

Hogyan kell értelmeznünk ezt a nem egész számot? Hogyan lehet csak egész számú kérdést feltenni? — Sehogy.

Ha csak *egy* ilyen választási esetet vizsgálunk, akkor a felteendő kérdések minimális száma a H értékhez legközelebb eső, H-nál nagyobb egész szám. Esetünkben 2. De ha ezt a választást igen sokszor megismételjük egymás után, akkor kimutatható, hogy a felteendő kérdések *átlagos* száma a választások számának növekedésével H értékéhez tetszőlegesen közelíthető.

Mi van akkor, ha az egyes üzenetek (az egyes állapotok) valószínűsége nem egyenlő: például a 32 jelölt közül az egyik esélye 50 százalék, a fennmaradó 50 százalékon pedig a többi 31 jelölt osztozkodik? Vajon ilyen esetben is 5 bitnyi bizonytalanságot szüntetett-e meg (hordoz) az a közlés, hogy ki az elnök? Sejthetjük: nem, — nyilvánvalóan kevesebbet. Szélső esetben, ha lenne például egy teljesen biztos, 100 százalékos jelölt, akkor a „választás" eredményének közlése nulla bizonytalanságot szüntetne meg, a közlésnek ekkor nem lenne információtartalma.

Általános esetben, amikor az egyes üzenetek nem egyformán valószínűek, az ún. Shannon-féle formula segítségével számíthatjuk ki az információforrás (a helyzet) rendezetlenségére jellemző értéket:

$$H = p_1 \log_2 \frac{1}{p_1} + p_2 \log_2 \frac{1}{p_2} + \ldots p_N \log_2 \frac{1}{p_N}$$

N a lehetséges üzenetek száma, p_i pedig az i-edik üzenet valószínűsége. Akik arra kíváncsiak, hogy miért pont ezzel a képlettel fejezzük ki a bizonytalanságot, azoknak ismét az irodalmat — különösen Shannon gondolatmenetét — ajánljuk.

Ha két jelöltünk van, és az egyik megválasztásának valószínűsége $p_1 = 0{,}8$ (80%), a másiké $p_2 = 1 - p_1 = 0{,}2$ (20%), akkor a Shannon-féle formulával ezt kapjuk:

$$H = 0{,}8 \log_2 \frac{1}{0{,}8} + 0{,}2 \log_2 \frac{1}{0{,}2} = 0{,}72 \text{ bit.}$$

Amikor egy ilyen rendszerű választás után közlik az eredményt, nem 1 bit bizonytalanságot szüntetnek meg, hanem kevesebbet. Pongyolán fogalmazva: ilyenkor a közlés információtartalma csak 0,72 bit.

Nézzük meg most, hogyan alakul a kétjelöltes választási helyzet, ha az egyik jelölt választási valószínűsége 0-tól 100 százalékig nő (lásd a Shannon és Weaver könyvéből vett *ábrát*). A helyzet rendezetlenségének maximuma az egyenlő esélyű helyzetnél van.

Közbevetőleg megjegyezzük: Shannon és Weaver is felhívta a figyelmet arra, hogy „azok számára, akik fizikát tanultak, feltűnik, hogy az . . . információ mértékeként *entrópia*szerű kifejezés jelenik meg. Az . . . entrópia szorosan kötődik Bolzmann nevéhez . . ." és a termodinamikához. Az információforrás rendezetlenségének mérőszámát tehát ugyanolyan képlettel számíthatjuk ki, mint a termodinamikában az *entrópia* mértékét. Ezért az entrópia fogalmát az információelmélet is átvette.

Hasznos dolog az információforrás rendezetlenségének (entrópiájának) mértékét az entrópia lehetséges maximális értékéhez viszonyítani. Ilyenkor kapjuk az ún. *relatív entrópiát:*

$$\text{relatív entrópia} = \frac{H}{H_{max}} = \frac{H}{\log_2 n}$$

ahol *n* a lehetséges üzenetek száma. Itt csak megemlítjük — de a továbbiakban nem használjuk fel a gondolatot —, hogy gyakran használják még az ún. *redundancia* fogalmát, amely definíciószerűen:

$$\text{redundancia} = 1 - (\text{relatív entrópia}).$$

Egy olyan üzenetkészlet relatív entrópiája, melyben minden üzenet egyformán valószínű: 1, a maximális érték. Ebben az esetben a redundancia 0.

A fentiek alapján azt is mondhatjuk, hogy az entrópia az egy üzenetre eső átlagos információ értéke. Ha olyan speciális üzenetrendszert (jelrendszert) használunk, melynek mindössze *két* eleme van (jelöljük az egyiket 0-val, a másikat 1-gyel), akkor ennek a jelrendszernek a maximális entrópiája:

$$H_{max} = \log_2 2 = 1.$$

Ha egy 32 üzenetből álló üzenetrendszert (például egy ábécét) le kívánnánk „fordítani" erre a két elemből álló „nyelvre", akkor a $\log_2 32 = 5$ értékből következően egy eredeti üzenethez (az ábécé egy-egy betűjéhez) 5 darab kételemű (bináris) jelsorozatra lenne szükségünk. (Az ilyen típusú hozzárendeléseket kódolásnak nevezzük. Az ellentétes irányú folyamat a dekódolás.)

Számítógépeinkben az információt itt most nem részletezett gyakorlati okokból mikroszkópos méretű építőelemek *két* fizikai (elektromos, elektronikus) állapotának formájában tároljuk (számítógépeink ma tipikusan bináris működésűek.) Számítógépeink-

egy magyarul közel 40 éves késéssel kiadott klasszikus művet: Claude Shannon — Warren Weaver: A kommunikáció matematikai elmélete (OMIKK, 1986.). Lehet, hogy ez riasztó cím, de ha érdekli a magazin olvasóit is, hogyan mérik az információ mennyiségét, beszámolónkban talál figyelemre méltó szempontokat. Írásunkban segítségül hívtuk még Rényi Alfréd: Napló az információelméletről (Gondolat, 1976.), Andor Csaba: Jel — kultúra — kommunikáció (Gondolat, 1980.) és Fényes Imre: Entrópia (Gondolat, 1962.) korábban megjelent érdekes könyvét.

ben a velük megoldandó feladatok „világa" az építőelemek halmazának e két fizikai állapotú „mintázataiban" tükröződik. Az információ *mennyiségének* a Shannon(—Hartley)-formulával kiszámítható mennyisége úgy válik kézzelfogható gyakorlati jelentőségűvé a számítástechnikusok részére, hogy megadja: a tár hány kétállapotú elemét kell biztosítani egy-egy üzenet/jelrendszer egy-egy elemének tárolásához.

Ha például a 0-tól 65535-ig terjedő, összesen 65536 darab egész számnak (integernek) megfelelő jelet (üzenetet) akarjuk a „0"/„1" nyelvre lefordítani, akkor ehhez $\log_2 65536 = 16$ darab kétállapotú tárolóelemre van szükség.

Ha az üzenetkészlet például a naptári hét napjaiból (hétfőből, keddből stb.) áll, akkor ennek bináris kódolásához $\log_2 7 \cong 2{,}8$ bitre lenne szükségünk. (Persze ha a hét napjait mint szavakat a magyar ábécé szerint kódolva tárolnánk, akkor többre. Például betűnként 5 bittel számolva a „vasárnap" nyolc betűből álló szó esetén $5 \times 8 = 40$-re.) Mivel törtszámú kétállapotú tárolóelemnek nincs értelme, így a naptári hét napjait némi „pazarlással" 3 darab kétállapotú elemben tükröztethetjük a gépben. Például így:

000 hétfő
001 kedd
010 szerda
011 csütörtök
100 péntek
101 szombat
110 vasárnap
111 *kihasználatlan kód* (állapot)!

Amikor egyes programozási nyelvekben a *változók* ún. *típusáról* beszélünk, akkor voltaképpen arról van szó, mekkora a változók lehetséges *értékkészlete*. Mivel a változók (is) a számítógéppel megoldandó feladatok világának részei (elkülönített objektumai, illetőleg azok tulajdonságai) helyett állnak, úgy is mondhatjuk, hogy a változók értékkészletét az szabja meg, hogy a *külvilág* hányféle dolga, megkülönböztetett állapota, tulajdonsága helyett állhatnak.

Nézzünk például egy „állapot" nevű változót egy Pascal nyelvű programban:

VAR állapot: (jó, használt, kopott, szakadt);

ahol a zárójelben — a nyelv szabályai szerint — a változó típusára jellemző értékkészletet soroltuk fel. (Az „állapot" nevű változó ezekből az értékekből vehet fel — természetesen mindig csak egyet-egyet). Egy ilyen változó számára a tárban legalább

$$\log_2 4 = 2,$$

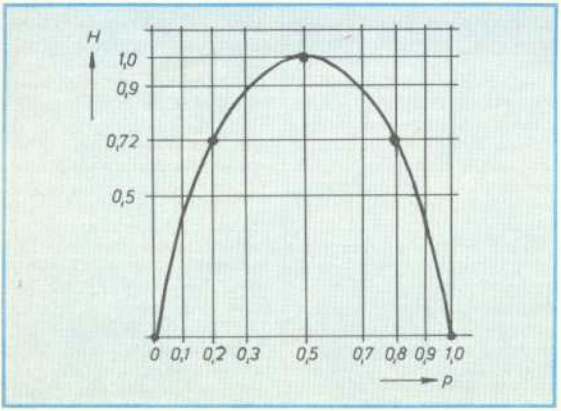
azaz legalább két darab kétállapotú elem elkülönítése/fenntartása szükséges. Ezt a tárterülettel való gazdálkodás szempontjából jó előre tudni. A „korszerűbb" nyelvek többek között ezért *is* írják elő a változók típusának explicit deklarálását (megadását).

Ide kívánkozik egy elégszer nem ismételhető megjegyzés: ha a gép például a „kopott" értékhez hozzá is rendeli mondjuk a „10" bináris kódot, azt, hogy ez mit jelent, csak a mi fejünkben (a számítógépet használó, azt programozó emberek fejében) kapcsolódik össze egy tényleges, például a Patyolatba leadott ruha konkrét tulajdonságával. A gép erről a hozzárendelésről semmit sem tud! A feladatok világának és a gép kétállapotú elemeinek mintázásában tükröződő szimbólumok világának összekapcsolása a géppel feladatot megoldó ember *fejében* történik.

Még valami: egy-egy jelrendszer maximálisnál kisebb entrópiáját rendszerint nem tudjuk kihasználni, amikor a jelrendszert binárisan kódoljuk a tár számára. A ma használatos gépek tára tipikusan 8 darab kétállapotú elemet tartalmazó ún. bájtokra van felosztva. A 8 darab kétállopatú elem $2^8 = 256$ darabból álló jelkészlet kódolására alkalmas. Ennyi lehetőség elegendő a világ legelterjedtebb ábécéinek (kis- és nagybetűk, írásjelek, számjegyek, különleges, ún. vezérlőkarakterek) együttes befogadására. Tudjuk, hogy az egyes betűk előfordulásának valószínűsége sem a magyar nyelvben, sem más nyelvekben nem egyforma: a jelrendszer entrópiája kisebb a maximálisnál. Ezért „trükkös kódolással", egyes esetekben (!) lehetséges lenne egy 256 jelből álló ilyen jelrendszer kódolása 8-nál kevesebb bitben is. A „trükközés" azonban egyrészt *általában* nem lenne végrehajtható, másrészt sokkal többe kerülne, mint amibe a jelenlegi „pazarlás" kerül.

— KE —

Két, p és (1—p) valószínűségű lehetőség entrópiája



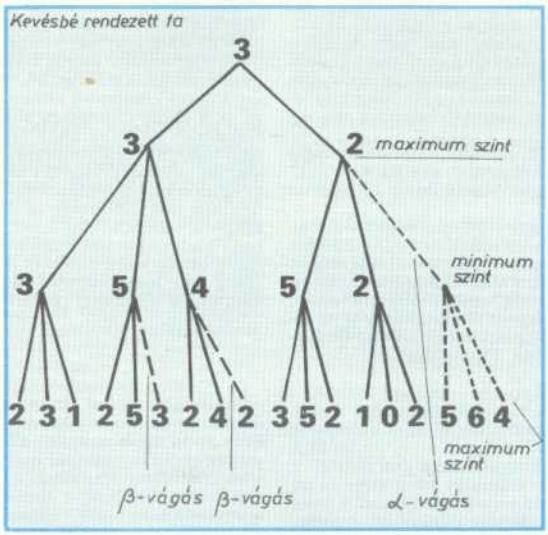

```
HELYREIGAZÍTÁS

Lapunk előző számában az Olvasunk... című cikkhez tartozó progr
am sajnálatos módon hibásan jelent meg. Olvasóink elnézését kér-
jük, és közöljük a javított listát.


PROGRAM verem-tár (input, output);
CONST  N= <a verem-tár mérete> ;
VAR    nemüres, nincstele : BOOLEAN;  n:  0..N ;
       verem : ARRAY [0..N-1] OF INTEGER;
       x,y   : INTEGER;
 :
PROCEDURE elvermel;
 BEGIN
  IF  n<N  THEN BEGIN
   verem [n] :=x; n:=n+1;  nincstele:= n<N;
   nemüres:=TRUE;
                 END
END;
PROCEDURE kivesz;
 BEGIN
   IF  n>0  THEN BEGIN
   n:= n-1; x:=verem [n]; nemüres:=n>0; nincstele:=TRUE;
                 END
 END;
    :
 BEGIN (* A  főprogram kezdete *)
    :
   n:=0; nemüres:= FALSE; nincstele:=TRUE;
    :
   IF  nincstele  THEN BEGIN
       X:=  <az "elvermelendő"érték> ;
       elvermel (* procedure-hívás *) END;
    :
   IF  nemüres  THEN BEGIN
       hívnaz; (* procedure-hívás *)
       y:=x                END;
    :
END.(* A főprogram vége *)
```

A játékfa és kiértékelése 2.

Az előző alkalommal egy programot mutattam be a játék fájának kiértékelésére. Láthattuk, hogyan kell a minimax algoritmust beprogramozni, és ez hogyan gyorsítható az alfa-béta eljárással. A következőkben az algoritmus gyorsításának további módjával foglalkozom.

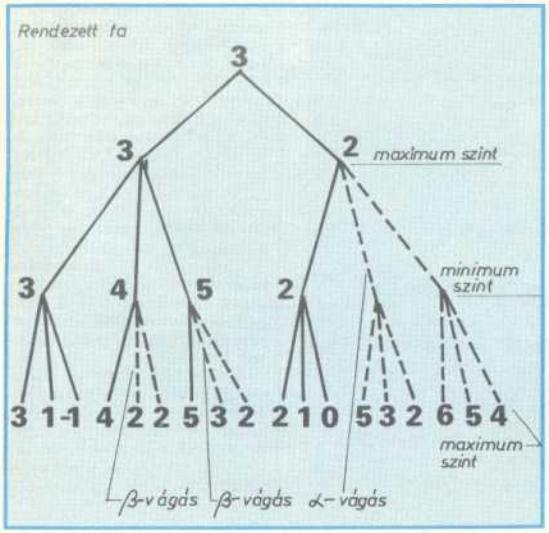
Először is nézzük, hogy az alfa-béta algoritmus alkalmazása mennyire hatékony.

Egy d féllépés mélységű játékfán, ahol minden levélnek átlagosan b ága van, b^d számú végállás jön létre. A minimax algoritmus b^d végcsomópontot értékel. Az alfa-béta algoritmus alkalmazásával ez kedvező esetben $2 \times \sqrt{b^d}$-re redukálható, ha mindig az illető levélben a legjobb lépést vizsgáljuk meg először.



Ha egy *hat féllépés mélységű* játékfát vizsgálunk és figyelembe vesszük, hogy általában egy állásban a megtehető lépések átlagos száma 38, akkor a minimax algoritmus alkalmazásával $38^6 = 3 \times 10^9$ állást kellene átvizsgálni, míg az alfa-béta eljárással optimális esetben „csak" $2 \times \sqrt{(3 \times 10^9)} = 109\,744$ végállást. Még ez is nagy szám, pedig csak optimális esetben ennyi, vagyis ha a fa rendezett; egyébként a megvizsgálandó végállások száma jóval több. Ez a szám a gyakorlatban általában egy nagyságrenddel nagyobb, ezért egy gyors számítógép esetében is, amely másodpercenként ezer állást tud értékelni, túlságosan is hosszúra

```
AlfaBeta(p: pozicio; alfa,beta,melyseg: integer): integer;
var
    ertek,sz,t,m : integer;
    lepesek : array[1..MAX_SZELESSEG] of integer;

begin
    if melyseg = 0 then  { terminalis csomopont? }
        return( Pontertek(p) );
    sz := Lepesgenerator( lepesek );
    if sz = 0 then       { nincs legalis lepes? }
        return( Pontertek(p) );

    ertek := -∞;
    for m := 1 to sz do
        begin
            t := -AlfaBeta( p.lepes[m], -beta,-alfa,melyseg-1 );

            if t > ertek then
                ertek := t;

            if ertek >= beta then
                { Levagja a fa tobbi reszet }
                return( ertek );

            alfa := Max( alfa,ertek );
        end;

    return( ertek );
end.
```



(Minden szinten az illető fél legjobb lépése a fa legbaloldalibb ága)
Minél rendezettebb a játékfa, annál kevesebb lépést kell megvizsgálni, így mélyebb elemzésre jut idő

nyúlna a parti. Lépésenként átlagosan 20 percet „gondolkodna" a számítógép. Ezért feltétlenül szükséges az algoritmus hatékonyabbá tétele.

Amint már említettem, a fenti számadatok a játékfa értékelésének optimális esetére vonatkoznak, ami a fa teljes rendezettségét követeli meg. Ha ez nem így volna, már az értékelés előtt tudnánk a legjobb lépést, és az állásértékelésre nem is lenne szükség. Viszont elvégezhetünk egy előzetes értékelést, amely csak hozzávetőlegesen számol, és ennek segítségével bizonyos fokig rendezhetjük a játékfát.

Vigyázni kell arra, hogy minél több szempontot veszünk figyelembe az előre történő értékelésnél, annál több időbe kerül a fa rendezése, és végül az algoritmus lassabb lesz, mint előre rendezés nélkül. Ugyanis, ha a fa előre rendező algoritmusa nagyon precíz, akkor az értékelés a minimax algoritmushoz fog hasonlítani. Nagyon nehéz megtalálni az optimális középutat: azt, amelyik viszonylag eléggé pontos, bizonyos mértékig megbízhatóan rendezi a fát, de azért gyors, mert (és ezért) kevés szempontot vesz figyelembe.

A fa előre rendezésére használhatjuk még az ún. iterációs eljárást, amelynek szintén az alfa-béta értékelés az alapja, de nemcsak a fa végpontjában értékel (például a hatodik félépés szintjén), hanem a játékfa közepén is, és ennek alapján rendezi a fát. Ha a játékfa végpontjaiban csak egy levél értékelését takarítottuk meg, akkor a fa közepének értékelésével már nem vesztettünk semmit. A legtöbb esetben ez az előre rendezés elég pontos és gyors, ezért a jelenlegi sakkprogramok legtöbbje ezt alkalmazza.

Végül a listán bemutatjuk, hogyan is lehet az iterációs eljárást programnyelvre ültetni.

KOVÁCS P. ATTILA

GÉPI ELLENFELEINK

Az Elite család

A Hegener+Glaser Mephistói után ez évi 1. számunkban megkezdtük a Fidelity gyártmányok ismertetését. Kétségtelen, hogy mind minőségben, mind választékban e két cég képviseli napjainkban a világ élvonalát. Ezúttal Fidelity újabb és legújabb készülékeit ismertetjük.

Gyártók és programozók

Mielőtt a készülékekről szólnánk, ezen a ponton teszünk egy kis kitérőt.

A Mephistók ismertetéséből is észrevehette az olvasó, hogy egy-egy új típus megjelenése a gyártó és a programozó együttes elhatározásától függ. A döntő szó természetesen a gyártóé, de igen sok múlik azon, hogy milyen jellegű a kapcsolata a programozóval, illetve a programozókkal, ha valaki többet foglalkoztat. Kié a kezdeményezés? Milyen meggondolásokból születik meg egy új készülék? Ebből a szempontból igen nagy különbség van a Mephistók NSZK-beli gyártójának igazgatója, *Manfred Hegener* és programozói, illetve az USA-beli Fidelity-főnök, *Sidney Samole* és a Spracklen házaspár kapcsolata között.

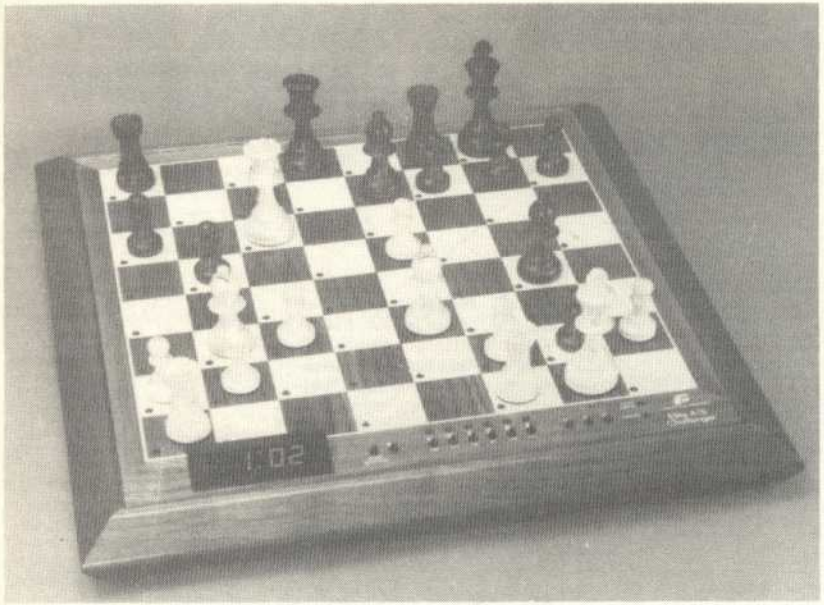
A programozó sohasem a gyártó alkalmazottja, hanem önálló személy, illetve leggyakrabban saját cége van, és a gyár szerződéses viszonyban áll vele. Hegenerék — amint megírtuk — több programozóval vannak kapcsolatban, és elvben bárkitől megvesznek új elgondolásokat tartalmazó, jó programot. Fidelity részére ezzel szemben immár több mint hat éve Kathe és Dan Spracklen készítik minden készülék programját, s a házaspár változatlanul a világ legjobbjai közé számít. A programozóknak a hardverek adottságaihoz, a gyár fejlesztési koncepcióihoz alkalmazkodniuk kell ugyan, Spracklenék azonban beleszólnak ebbe, és végül is az új készülékek általában közös megállapodás alapján születnek meg.

A készülékek vásárlóit ez annyiban érdekli, hogy a hardverek fejlesztésétől — ami egyrészt a műszaki paraméterek javításában, másrészt a készülékek könnyebb kezelésében, több információ kiírásában, s a kivitel esztétikájában nyilvánul meg — nem szenved-e a moduláris rendszer. Fidelitynél ez örök probléma. A cég üzleti stratégiája kezdetben az volt, hogy amikor Spracklenék új programot írtak vagy jelentős továbbfejlesztést hajtottak végre, akkor új készülékkel jelentek meg — ebben jobb üzletet láttak —, ez azonban a moduláris rendszer elterjedésével a visszájára fordult. A vevők nem akartak mindig új és új készüléket vásárolni.

Fennállása óta legalább két alkalommal Fidelitynek — készülékeinek, programjainak kimondottan magas színvonala ellenére — komoly pénzügyi nehézségekkel kellett megküzdenie. Aligha tévedünk, ha úgy gondoljuk, ebben a piaci feltételek változása játszott erősen közre: újdonságaikkal nem elég olcsón jöttek a piacra. De mindig talpraálltak, és most is az élen vannak.

Elite és megint Elite

Az első Elite A/S (Auto-Sensory) jelzéssel 1982-ben jelent meg, és a travemündei 2. mikroszámítógép világbajnokságon óriási fölényű Fidelity X készülék programját tartalmazta, javított megnyitási könyvtárral. 3 MHz sebességen működött. Igen szép fatábla és finoman kidolgozott készlet, al-

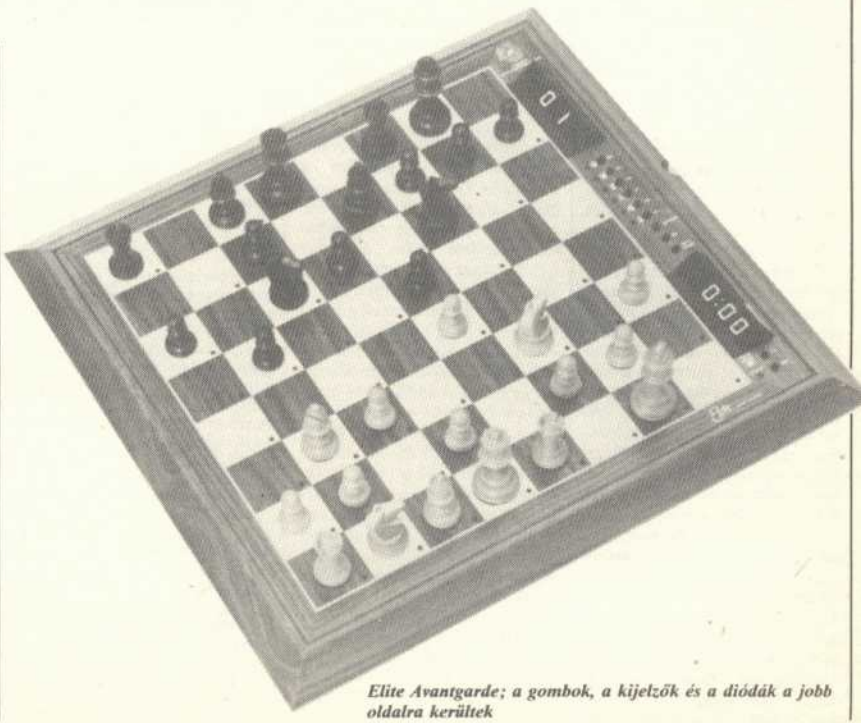


Elite A/S, a budapesti világbajnok

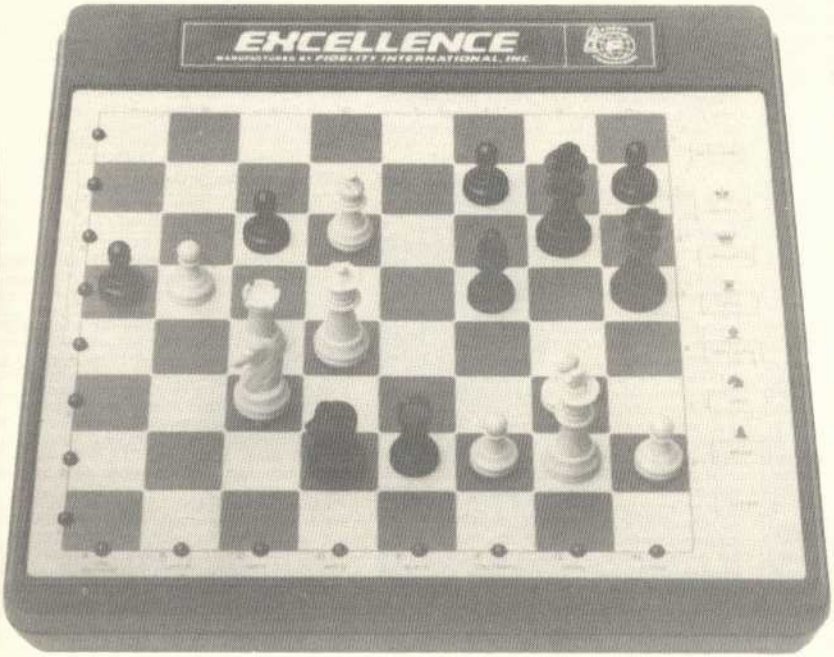
só nyomógombokkal és kijelzővel alkotta a hardvert, a tábla mérete 46 × 48 × 4,5 cm volt, illetve ezek az adatok, amint hamarosan kiderül, nagyrészt ma is érvényesek. Ugyanakkor elkészült a gyárban a luxus kivitelű Prestige, ugyanazzal a programmal, és nemcsak sokkal szebb, hanem egyben hatékonyabb hardverrel is. A processzor 4 MHz-en futott, a tábla mérete meghaladta az Elite-ét, túl volt az 50 × 50 cm-en és az 5 cm-es magasságon. Az Elite ára 1000, a Prestige-é 1500 dollár körül volt, ami természetesen igen magas. De már akkor tudtuk, hogy ezek a típusok, illetve programok nem maradnak soká a piacon. A gyár és Spracklenék a 3. budapesti vb-re készültek, és kérdésünkre közölték, hogy ismét teljesen (ami legalábbis ennyit jelent: nagymértékben) átdolgozzák programjukat, továbbá kijelzőjük minden eddigi készüléknél több információt fog nyújtani a kezelőjének.

1983 októberében, a budapesti vb-n érdekes dolog történt. Elite A/S — még mint kísérleti készülék — a hét játszmából elért 6 pontjával egy teljes egységnyi előnnyel megnyerte a világbajnokságot — a legjobb Mephisto (amely akkor a III. program javított és gyorsított változata volt) és két kísérleti Novag készülék előtt. A Prestige további fél ponttal hátrább, az ötödik helyre került. Ez annál feltűnőbb volt, mert a kettő ugyanazzal a programmal futott, csak az Elite 3, a Prestige pedig 4 MHz-en. Újabb tanulságul szolgált az eset arra nézve, hogy a működési sebesség kismértékű emelése alig befolyásolja a játékerőt. David Levy nemrég kimutatta, hogy a vizsgálati mélységnek „brute force" útján egy fél lépéssel való növeléséhez — azaz vagy egy világos, vagy egy sötét lépéssel — a sebességet öt-hatszorosára kell fokozni! Jó, ha a számítógépek vásárlói gondolnak erre, amikor két készülék összehasonlításánál a MHz-ből akarnak kiindulni.

Az új Elite—Prestige páros valóban kiválónak bizonyult. A programot moduláris rendszerrel cserélni lehetett, megnyitástárakat is be lehetett iktatni. 50 százalékos eredményt ért el a nagyszámítógépek kö-



Elite Avantgarde; a gombok, a kijelzők és a diódák a jobb oldalra kerültek



Excellence; a lépéseket koordinátákban jelzik

zött a New York-i vb-n, és 1983—1984-ben vitathatatlanul a világ legerősebb sakkszámítógépének tekintették. Csak az ára volt nagyon magas, különösen, ha a dollár akkori értékére gondolunk. Közölték is, hogy a Prestige nem kerül sorozatgyártásra. Viszont forgalomba hozták olcsóbb, műanyag kivitellel a Sensory 12-est, amely az Elite programját tartalmazta, s ez már 800 DM körüli áron kapható volt. Ma is megér 500—600 DM-et; noha a Mephistók és a Scisys vagy a Novag korszerűbb készülékei azóta utólérték, elhagyták.

Problémát okozott azonban, hogy az Elite A/S nem kapott új nevet, és csak a gépek gyártási számából lehetett megállapítani, hogy a régi vagy az új, budapesti program van-e bennük. Új modul feltétlenül legalább megjelölést kíván!

Ez a gond csak fokozódott 1984—1985-ben. Spracklenék ugyanis ismét továbbfejlesztették programjukat, s a glasgow-i 4. vb-n holtversenyben megint az élre kerültek. Ekkor kezdték a hirdetésekben és árusításkor közölni, hogy a budapesti vagy a glasgow-i programról van-e szó. Azt már csak valószínűsíteni lehet, hogy a glasgow-i programnak is kijöttek egy további javításával, mert noha hirdették, csak egyes országokban látták el a berendezéseket I. II. és III. jellel. Az eredeti, régi Elite-ek közben kifutottak, de amikor alkalmi vételként kínálják valamelyiket, bizony alig lehet megállapítani, hogy tulajdonképpen melyik Elite-ről van szó. Aki használt Elite-et vásárol, okvetlenül forduljon szakértőhöz! Hiszen hardverjük azonos, valamennyire „Elite A/S Challenger" van írva, fehér betűkkel.

Avantgarde, Excellence

A homályt csak fokozta, hogy Amszterdamban az 1985. évi vb-n — fennállása óta először — a Fidelity nem vett részt. Pedig közben már megint új programon dolgoztak az ötletekben kifogyhatatlan Spracklenék. Egy kísérleti készülékkel megnyertek Amszterdam előtt egy versenyt Moblieban, s röviddel utána a piacon is megjelent Fidelity a Mobliebe-program kereskedelmi verziójával, a legújabb Elite Avantgarde-dal! Ennek mérete azonos a régi Elite-évei, de most már új hardverje van, két kijelzővel, ami természetesen azt jelenti, hogy még több szolgáltatást nyújt. A különbséget képeink mutatják. A készülék kiváló, számos tesztelő egyöntetű véleménye szerint játékerőben megközelíti a Mephisto Amsterdamot, s kezelése sok tekintetben praktikusabb azénál. Ez nagy szó, mert processzora változatlanul 8 bites. Ára pedig olcsóbb annál: eredetileg 650 dollárért hirdették, de már most kapható például hangjel nélkül 500 dollár alatt. Az NSZK-ban 1300 DM körüli áron kínálják. Természetesen ez még mindig nagy pénz, csak a legigényesebb vásárlók fizethetik meg.

Kihozott azonban Fidelity az Avantgarde-dal egy időben egy viszonylag tényleg olcsó, műanyag készüléket is, az Excellence-t, amely az elmúlt hónapokban a tesztelők ranglistáján igen előkelő helyre küzdötte fel magát. Az NSZK-ban 400 DM körül árulják. S nem sokkal utána megjelent a Par Excellence is, amely nem más, mint az Avantgarde egyszerűbb, műanyag kivitelű és csak szolgáltatásait tekintve szerényebb testvére; 750 DM-ért hirdetik, s ugyanakkor az Excellence árát nem egy helyen 350 DM-re szállították le!

Mindezeket a készülékeket magunk is kipróbáltuk, és épp oly melegen ajánlhatjuk a szerény devizával rendelkező magyar vásárlóknak, mint a hasonló árfekvésű, új Mephisto típusokat.

LINDNER LÁSZLÓ

Minden hétfőn 17-től 19 óráig

a Mikroszámítógép Magazin munkatársai és felkért szakértők válaszolnak az olvasók kérdéseire a szerkesztőségben: Budapest, II. ker. Fő u. 68. I. em. 109-ben, vagy a 154-090 és a 154-250-es telefonon.
(Az első találkozási alkalom: szept. 7.)

Minden kedden 17-től 20 óráig
ENTERPRISE-klub
a VSZM Közösségi Házban
(Bp. XI., Fehérvári út 120.)
Klubvezető: Romvári Gábor

Amikor ez a szám megjelenik, már véget ért a vakáció, megkezdődött az iskolaév; a nyári szabadságokat is többnyire letudtuk. Olvasóink bizonyára sok ötletet és valószínűleg számítógépeket is hoztak a nyaralásból, különösen, ha szabadságukat külföldön töltötték. Várjuk élménybeszámolóikat, javaslataikat, amelyeket igyekszünk hasznosítani.

Ifj. Gulyás Antal, Szeged,

Török u. 3. 6722

A Mi és a számítógép c. sorozatban hallottam, hogy az Enterprise számítógépre Spectrum-emulátor készül. Szeretném megtudni, hogy mikor jelenik meg, mennyi lesz az ára, tartalmazza-e 100 százalékban a Spectrum ROM-ot, és hogy egyezik-e a Spectrum „furcsa" video-RAM elosztásával, színkezelésével. A másik kérésem: ha van rá mód, küldjék el dr. Ada-Winter Péter—Ada-Winter Dávid: A ZX-Spectrum c. könyvét. Utánvéttel megvenném, ugyanis hónapok óta hiába keresem; az egyik szakboltban nem is hallottak róla.

Az Enterprise-ról viszonylag kevés információ jelent meg. Akik megvették ezt a gépet, azt mondják, hogy jó, gyors, megbízható, különösen a beépített szövegszerkesztőt dicsérik. A Spectrummal azonban még nem kompatibilis. Meg nem erősített hírek keringenek, hogy lesz hozzá Spectrum-emulátor, de az Enterprise mégsem Spectrum.

Más olvasóinknak is üzenem, hogy könyvek, alkatrészek, számítógépek, perifériák és más hasonló cikkek vásárlására és elküldésére a szerkesztőségünk nem vállalkozik.

Dr. Bangha József, Budapest,

Bartók Béla út 50. 1111

Három kisunokámmal együtt olvasom és használom lapjukat. Most azonban mind a négyen csődöt mondtunk, és segítségüket kérjük! Először előfordult már, hogy a lapjukból begépelt program néhány jele, sőt sora is olvashatatlan volt, de eddig megbirkóztunk ezekkel a problémákkal. Sajnos olyan kevés az unokáim C16-osára írt programa, ami közvetlenül begépelhető, hogy az átírásokat is meg kell kockáztatnunk. A májusi számban azért örültünk különösen a 38. oldalon a KE jelzésű cikkben közölt „Huszár a sakktáblán" programnak, mert igen részletes leírás volt hozzá, ami kellett is, mert a program sok betűjét még nagyítóval is nehéz volt felismerni. A probléma a 9310—9320-as sorokban megoldódott a kettős zárójelek elhagyásával, de a 11310—11350-es sorokat nem fogadja el a gép, még akkor sem, ha a kis hibát kijavítjuk, hanem megrögzötten adja a bad subscript jelzést.

KE és a nagypapa telefonon tisztázták az ügyet, de miután másokat is érdekelhet a javítás, közöljük KE válaszát.

„Egy száznál több sorból álló program beírása és felélesztése nem könnyű feladat. A nagypapa és három kisunokája kemény fába vágta fejszéjét, de — mi is úgy gondoljuk — megéri. Az egyes gépek BASIC „tájszólásai" (különösen a finom részletekben) eltérnek. Itt például az lehet a probléma valószínű oka, hogy a C16-os kis méretei miatt a tábla méretét az 1800-as sorban (DIM A(8,8)) le kellett csökkentenünk kisebbre, például A (5,5)-re; a PRINT utasításban „kifutnak" az index- (subscript-) tartományból.

Mivel nem önök az elsők, akik ezt a programot otthon életre kelteték (igaz, mások ezt C64-en és nem C16-on tették) és segítségre szorultak, a szerző ismeri és átélte a „honosítás" (hogy ne mondja: „otthonosítás") nehézségeit."

Nagy Tibor, Miskolc,

Árpád út 48. 2/1. 3534

A Rádiótechnika 1987. februári számában olvastam egy hirdetést — ezt küldöm el levelemben —, melyben egy illető pénzért kínált programokat. Egy kissé dühített a dolog, mert biztosra vettem, hogy nem saját készítésű, hanem külföldi programokat árusít (FORTH, HEADER, JUSTAGE, gyorsítók és monitorok). Eszembe jutott az ön cikke, amely az 1985/2. Mikromagazinban jelent meg Copyright címmel. Gondoltam, talán az illető majd magától rájön, hogy nem tisztességes, amit csinál, és jobb útra tér. Ekkor ért a keserű meglepetés. Ugyanez a személy az 1987/6. Mikromagazinban hirdetett, és ugyanazokat a programokat, ugyanúgy, pénzért kínálta. Remélem, nem sikerül neki! Nem lehetne fellépni az ilyen emberekkel szemben, esetleg megbüntetni őket? Az ehhez hasonló hirdetéseket pedig kiszűrni a Mikromagazinból és máshonnan is. Úgy érzem, károsan hatna a programcsere-mozgalomra, ha ez a dolog elterjedne.

Azt hiszem, hogy igaza is van, meg nincs is. A hirdetéseket elolvasva ugyanis nekem nem az jutott az eszembe, hogy feladójuk lopta, esetleg jogtalanul árusítja a programokat; arra gondoltam inkább, hogy P. Gy. Szilasligetről egy nagyon ügyes szoftverfejlesztő, esetleg van egy gmk-ja is, amely rendszereket készít, és ezeket ajánlja a vevőknek. El sem tudom képzelni, hogy Copyright cikkem után bárki is megpróbálkozna mások szellemi termékének jogtalan árusításával. Legalább röstelkednie illene.

Ami a µM hatósági jogkörét illeti, ma még nem rendelkezünk olyan lehetőségekkel, hogy a lapban ajánlott programok szerzőségét vizsgáljuk. Ha jól tudom, szerzői jogi kérdésekben a sértett polgári pert indíthat, a szerzői jogvédő ebben támogatja, és ha igaza bebizonyosodik, a bíróság a kártérítést megítéli.

Jakab Szilárd, Csorna,

Petőfi-ltp. 7. 9300

Az újságjukat böngészve mit lát az ember: Commodore programok, tanácsok egymás hegyén-hátán. És mit tegyen az, akinek ennél kisebb gépe van? Én például egy 32 k RAM-os ZX81-gyel dolgozom. Szerintem nagyon frappáns ötlet lenne a következőket megvalósítani: 1. minden számban külön rovatot szerkeszteni a hazánkban legelterjedtebb gépeknek, és nem összevissza közölni róluk információkat; 2. közölni a jelenlegi gép- és programárfolyamokat; 3. az újság végére különböző felhasználói táblázatok kerülhetnének.

Azt hisszük, hogy a µM-ban közölt programok kb. ugyanazt az eloszlást mutatják, mint az iskolai, illetve a házi számítógépek. Úgy látszik, hogy Commodore gépből van a legtöbb, ti. leginkább ilyen programokat kapunk. Beküldött ZX81-es program ritka, mint a fehér holló, de azt hiszem, hogy a gép is. A jelenlegi program- és gépárfolyamokat azért nem közöljük, mert a lap átfutási ideje elég hosszú, így ezek a „friss" információk megjelenésük idejére már el is avulnának.

Lilik Gábor, Fehérgyarmat,

Tömöttvár út 46. 4900

Városunkban nincs mikroklub, eddigi iskolámban pedig a számítógép nem volt (és nincs) a legjobb kezekben, így csak egy szűk réteg használhatja. Nekem van egy VC20 számítógépem, de sok számítógép-tulajdonos van még a városban. Talán klubot is lehetne alakítani, de erre mint gyerek, én nem vagyok képes. Persze nem ezért írtam önöknek. Lenne néhány észrevételem a laphoz. Például az 1987. évi 1. szám 19. oldalán a VC20 tárcimeit közölték; miért nincs ennek a cikknek folytatása mondjuk a C64-nél? Más. Nekem nagyon tetszenek Barna László cikkei, de nem értem, hogy ha a C16 nem ismeretlen többé, miért nem ír más gépekről ilyen „ismeretlen"-t?

Valamennyi javaslatát az illetékes rovatvezetőknek és szakíróknak továbbítottam. Ami a klubot illeti, azt javaslom, fogjanak össze, keressék meg az NJSZT Szabolcs-Szatmár megyei szervezetét, Nánási Erika elnökét (Megyei Tanács V. B.) vagy Simon Bélánét, a megyei szervezet titkárát (Mezőgazdasági Főiskola). Ők biztosan segíteni fognak a klub megalapításában.

Békési Gábor, Budapest

XVII., Kép Utcai Általános Iskola

Nemrégiben láttam a Personal Computer Worldben a SOFTSTRIP nevű, szerintem szenzációs adattárolási módszert.

Egy ilyen csikon a számítógép 5500 bájtot tud eltárolni, és ezt 30 másodperc alatt képes beolvasni, még gyűrött, maszatos papírról is. A közönséges, A/4-es lapra bármilyen mátrix-, illetve lézernyomtatóval kinyomtatható, fotokópia vagy fénymásolás útján sokszorosítható. Az adathordozó (papír) könnyen postázható, olcsó, és még egy csomó jó tulajdonsága van. Jó volna, ha lapjukból bővebb információkat kaphatnék erről a témáról. A mellékelt csík egyébként egy BASIC-ben írt rajzoló (sketch pad) program IBM PC-re, de beolvasható Apple Macintosh-ra is. Érdekesége, hogy az egymással nem kompatibilis gépeknél is lehetővé teszi az egymás közötti adatforgalmat.

Dr. Simonyi Endre válasza: A mellékelt csík az ún. bar code egy változata. Használatát többször ajánlották például a Byte c. szaklapban, még könyvet is adtak ki róla. A módszer az olvasó, az adathordozó problémái (például a szennyeződésérzékenység) miatt nem terjedt el.

A levelek nagyobbik részére szerkesztőink levélben küldtek választ. Itt csak a közérdeklődésre számottartó írásokat közöltük, melyeket ismételten megköszön:

KOVÁCS GYŐZŐ

Commodore C64 adatfeldolgozási lehetőségei (Budapest, 1986. IPIK, 136 oldal. Ára: 230,— Ft.)

A Commodore 64 személyi számítógép népszerűségével együtt nő a szoftverellátottsága és bővül az adatfeldolgozási programcsomagok, illetve az alapszoftverek száma is. Ezek azonban nem teszik lehetővé a speciális felhasználói igény kielégítését. A megoldást egy-egy rugalmasan alkalmazható rutingyűjtemény jelentheti.

Ez a könyv a fájlkezelési módszerek összefoglalását és a lehetséges bővítések ismertetését tartalmazza. Az elméleti módszereket példaprogramok teszik érthetőbbé. Mindezek együtt jól használható rutingyűjteményt alkotnak. A kötet elsősorban azoknak készült, akik már elsajátították a C64 alapfogalmait, és a gépet adatfeldolgozásra kívánják használni.

Futó István: CPC BASIC három szinten (Budapest, 1987. Műszaki Könyvkiadó, 383 oldal. Ára: 198,— Ft.)

Az utóbbi évek sikertörténeteinek egyik főszereplője az angol Amstrad cég, amely 1984-ben betört a számítógépgyártó óriások közé. A CPC 464, 664 és 6128-as gépcsalád, megtörve a többéves Commodore egyeduralmat, 1985-ben elnyerte az év számítógépe címet az otthoni gépek kategóriájában.

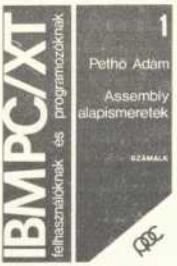
Mi a titka a CPC-k sikerének? Egy alapgép áráért teljes rendszert kap a vásárló.

A CPC interpretere a MICROSOFT BASIC mintájára készült, de alkotói néhány mesterfogással a kategória egyik leggyorsabb változatát hozták létre. A CPC BASIC-et nemcsak gyorsasága, hanem parancsokban való gazdagsága is jellemzi. A könyv célja, hogy a gép használóját végigvezesse a CPC BASIC rejtelmein. A három szintre való felosztás lehetővé teszi, hogy a különböző előképzettségű felhasználók tudásszintjük alapján találjanak érdeklődési körüknek megfelelő új ismereteket.

Az első szint a gép üzembe helyezésétől kezdve az első lépéseken keresztül elvezeti az olvasót a BASIC alapjainak elsajátításáig.

A második szint tartalmazza mindazt, amivel a CPC BASIC többet nyújt a hasonló kategóriájú gépek BASIC-jénél: grafika és hang, áttervezhető billentyűzet és karakterkészlet, multitasking és az ablakok programozása. A harmadik szint minőségi ugrás ez előzőkhöz képest.

A CPC gépi kódú programozásának lehetőségeivel ismerteti meg az olvasót. Kitér a CPC tárfelosztására, képernyőszervezésére, bemutatja a fontosabb rendszerrutinokat stb. A függelék mindhárom szinthez szolgáltat hasznos információt. Ábrákon és táblázatokon kívül a CPC BASIC összes kulcsszavának rövid ismertetése található benne.



Pethő Ádám: IBM PC/XT felhasználóknak és programozóknak 1. Assembly alapismeretek (Budapest, 1987. SZÁMALK, 223 oldal. Ára: 139,— Ft.)

A háromkötetesre tervezett sorozat elsősorban azoknak készült, akik már szereztek bizonyos programozási gyakorlatot, és IBM PC/XT személyi számítógéppel akarnak dolgozni. Ez az első kötet az assembler programozáshoz szükséges alapismereteket foglalja össze: a processzorismeretet, a fordító szolgáltatásait, a fordítás, a futtatható program létrehozásának menetét. A főbb fejezetek: Az Intel 8088 mikroprocesszor — Az Intel 8086/8088 címzési módjai, utasításkészlete — A MASM makroassembler szolgáltatásai — Az assembler és a LINK használata.

Tatchell, J. — Bennett, B.: Első könyvem a mikrókról (Budapest, 1987. Műszaki Könyvkiadó-Novotrade Rt., 47 oldal. Ára: 99,— Ft.)

Ez a szép kiállítású, színes képeskönyv gyermek és felnőtt kezdőknek mutatja be a mikroszámítógép működését, felhasználási lehetőségeit, és magyarázza meg azokat a szakkifejezéseket, amelyekkel a hasonló témájú könyvekben is találkoznak. Sok színes és fekete-fehér képpel illusztrálva ismertet meg a mikrók használatával, a feladatmegoldások módjaival.

A könyv a BASIC nyelvű programozás alapjainak tárgyalása után bemutatja a mikro működését és a szíliciumchipeket. Szól arról is, hogyan lehet összekapcsolni a mikrót az akár több ezer kilométerre levő nagyszámítógéppel. A mikrók kiegészítő eszközeiről: a fényceruzáról és a botkormányról is olvashatunk. Végül vásárlási tanácsadóban foglalják össze a szerzők a legismertebb házi számítógépek adatait, a mikrók legfontosabb jellemzőinek felsorolásával segítve a kezdő számítógép-használókat első mikrójuk megvásárlásában.

Pyle, I. C.: Az Ada programozási nyelv (Budapest, 1987. Műszaki Könyvkiadó, 310 oldal. Ára: 108,— Ft.)

Az Ada programozási nyelvet az USA hadügyminisztériumának támogatásával tervezték; nagy előrelépést jelentett a programozók igényeinek kielégítésében.

Az Ada beágyazott rendszerek programozására való: olyan rendszerekére, amelyekben a számítógép közvetlenül kapcsolódik ahhoz a berendezéshez, amelyet felügyel vagy irányít. A beágyazott rendszerek széles skálája az intelligens termináloktól a laboratóriumi adatfigyelésen, a számjegyvezérlésű gépeken, a navigációs és irányítási rendszereken, a kötegelt és folyamatos termelésirányításon, a mikroszámítógépet tartalmazó háztartási eszközökön át az automatizált gyártásig terjed.

A könyv hasznos lehet a programozást tanító tanárok és a programozási munkák irányítói számára egyaránt, mivel az Ada támogatja a nagyméretű programok kifejlesztését. A kötet — eredeti céljával összhangban — előzetes programozási ismereteket feltételez. Az egyes témaköröket a beágyazott rendszerek környezetében tárgyalja, olyan sorrendben, amely megfelel a programozásban szokásos gyakorlatnak.

Margaréta-kerekes nyomtató

A szocialista országok első margaréta-kerekes mikronyomtatóját 1985 őszén mutatták be Plovdivban, Evmolpia—100 néven. Ezt továbbfejlesztve, Izot 6311C néven kezdték az idén gyártani. Mindmáig Bulgária az egyetlen szocialista ország, ahol ilyen kategóriájú nyomtató készül. Az NDK-ban gyártanak ugyan margaréta-kerekes nyomtatókat, azonban a Robotron nyomtató minigépekhez nem illően robusztus, 30 kg-os; a kisebbek pedig billentyűzettel is rendelkező, de mikrogépekhez nyomtatóként is illeszthető elektronikus írógépek.

Az Izot 6311C típusú margaréta-kerekes mikronyomtató

Holográfia

A számítógéppel segített tervezés első céljai közé a gépkocsi-karosszéria tervezése tartozott. A számítógép azonban csak kétdimenziójú ábrákat tud készíteni. Ahhoz, hogy a készülő karosszéria tervét igazában térben láthassák, a tervezők kénytelenek annak minden egyes változatát agyagból vagy műanyagból megmintázni. A General Motors támogatásával a massachusettsi műegyetemen (MITI) olyan eljárást dolgoztak ki, amely — a holográfiát a számítástechnikával összekapcsolva — félkörben körüljárható hologramot készít a tervezett formáról.

A számítógép a tervezett kocsiszekrény adataiból először azt határozza meg, hogy az — mintegy ezer, egymástól egyforma szögtávolságban lévő pontból nézve — hogyan néz ki, és ezeket a képeket — a színárnyalatokat és az árnyékokat is figyelembe véve — egy 35 mm-es filmen rögzíti. Ezután minden egyes nézetről — mindegyikről a maga nézőpontjából — lézersugárral hologramot készít egy-egy 8,5 mm széles és 30 cm hosszú filmre, végül ezeket egyetlen filmen egyesíti. Ellentétben a hagyományos hologrammal, amelynek a hordozója (üveg vagy film) sík, ezt félhengerré hajlítják.

Amikor ezt a külön-külön nézetekből kialakított ezernyi — két kiterjedésű — hologramot lézerfénnyel megvilágítják, egységes térbeli forma hatását keltve a kocsiszekrény oly érzékletesen jelenik meg, hogy a néző úgy érzi: meg tudná fogni. A hagyományos hologram csak szűk szöghatárokon belül nézhető, s ha a fejünket e határokat túllépve mozdítjuk el, a kép eltűnik. Ez a félkörives hologram 180 fokos határokon belül nézhető, tehát félig körbejárható.

Ez a megoldás, bár olcsóbb és gyorsabb, mint a hagyományos formázás, még messze nem tökéletes. Egyrészt túlságosan munkaigényes: egy-egy ilyen hologramnak az elkészítése három kutatónak háromnapi munkájába kerül; másrészt pedig az eredmény sem teljesen tökéletes, hisz a hologramos karosszéria alá sem nézhetünk be s felülnézetből sem nézhetnek reá. Mégis komoly gyakorlati jelentőséget ad az eljárásnak az, hogy a tervezők a közbenső eredményeiket viszonylag könnyen megtekinthetik.

Gépi adathordozó papírból?

A Softrade angol cég fejlesztői készítettek egy olyan berendezést, amelynek segítségével a számítógépben tárolt programokat, adatokat tömöritett formában lehet kinyomtatni papírra, majd egy olvasó segítségével vissza is lehet olvasni. A tömörítés mintegy negyvenszeres, ami azt jelenti, hogy egy A/4 méretű lapra 40 oldalnyi információ kerül. A mintegy 200 angol fontba kerülő berendezés új távlatokat nyithat a már-már végleg eltűnt papíralapú gépi adathordozók terén. (Elterjedése esetén például az újságokban sem foglalnának el annyi helyet a közölt programok, és a fáradságos — és rengeteg hibával járó — bepötyögésük helyett pillanatok alatt beolvashatók lennének.)

További előnye e tárolási módszernek, hogy ezek a lapok közönséges másológéppel sokszorosíthatók, de a védelmük is megoldható: ugyanúgy, mint bizonyos szoftverdokumentációknál már találkozhatunk is vele, a speciális nyomdafestéket csak az emberi szem látja, a másológép nem, sötét pacnit képes csak róla csinálni. Az eljárás sikerét a hordozóanyagának, a papírnak a hétköznapi elterjedésével valószínűsíti. Pontosan ezért elsősorban könyvtári alkalmazására számítanak: a könyvtárakat minden bizonnyal ellátják a mikrofilmolvasóhoz hasonlóan ilyen olvasóberendezéssel is.

Társulás

Az Ipari Minisztériumhoz tartozó 110 könnyűipari vállalat közül jelenleg 63-nál alkalmaznak műszaki területen számítógépet. A vállalatok különösen a gyártás előkészítésénél látják ennek hasznát: a korábbi 2-3 hetes idő 1-2 napra rövidül.

Gondot okoz azonban, hogy a felhasznált számítógépek típusai és az alkalmazott módszerek igen eltérőek, így a társvállalatok nem tudják egymás sikeres szoftverfejlesztéseit átvenni, illetve egymás esetleges szabad kapacitásait kihasználni. Ráadásul a vállalatok nem eléggé érdekeltek abban, hogy a sikeres számítógépes megoldásokat egymásnak átadják, együttműködjenek.

A számítógépek szélesebb körű könnyűipari alkalmazására, illetőleg a viszonylag egységes géppark és munkamódszerek kialakítására az Ipari Minisztérium koordinálásával az idén gazdasági társulást hoznak létre. A közös szervezetben az egyes vállalatok és intézmények szakembereiből szakágazatonként munkacsoportok alakulnak, amelyek kidolgozzák és a társulás rendelkezésére bocsátják a legmodernebb számítógépes gyártáselőkészítési és termelési eljárásokat, munkamódszereket, programokat, és vállalják a már bevezetett rendszerek követését is.

A betegekért

A jövőben még biztonságosabbá válik a betegekkel kapcsolatos adminisztráció a Péterfy Sándor utcai Kórház-Rendelőintézetben, ugyanis az idén a már évek óta működő számítógépes rendszerhez a vérellátó, a gyógyszertár, valamint tíz kórházi osztály is hozzákapcsolódik. Az intézmény laboratóriumában már 1983 óta működik egy HT 680X típusú magyar számítógép, és a tapasztalatok bizonyítják, hogy alkalmazásával időt, vegyszereket, továbbá a vizsgálatokhoz azelőtt szükséges nagy mennyiségű papírmunkát lehet megtakarítani. A betegellátásban továbbra is keletkező rengeteg adat pedig rendezetté, könnyen áttekinthetővé válik.

Amikor valaki vizsgálatra jelentkezik a laboratóriumban, adatai a számítógépbe kerülnek, kap egy azonosítási számot, amely szerint a róla tárolt információk a későbbiek során bármikor visszakereshetők. A laboratóriumi elemzéseket végző speciális készülékek nagy része közvetlen kapcsolatban áll a számítógéppel, így a leletek többsége automatikusan a rendszerbe kerül. Az osztályokon elhelyezett terminálok segítségével az orvosok közvetlenül kérhetik a különböző vizsgálatok elvégzését, illetve visszakereshetik azok eredményét.

IIG-SCHATZKISTE
SCHÖNER WOHNEN

BTX-PRÄSENTATION
Bildgestaltung
Fotografie/Bewegung
Verwendung von
Laufbildern
ohne und mit
Farbwechsel.
Wasserhahn

0 « ZURÜCK
WEITER » 1

ÜBERSICHT 0
WEITER 1

Übersicht 0
Weiter 1

0 « ZURÜCK
WEITER » 1

Szki PRONET LOKÁLIS HÁLÓZAT

HARDVER

— A PROPER—16 (IBM kompatibilis) számítógépcsaládból összeállítható hálózat kétirányú busz szervezésű
— A hálózat kiépítéséhez PRONET lokális hálózati kártya is szükséges
— A hálózat állomásai közös átviteli közegre kapcsolódnak
— A hálózatba újabb állomások könnyen bekapcsolhatók

Főbb műszaki jellemzők

- adatátviteli sebesség: 1 Mbit/s
- adatátviteli közeg: csavart érpárú vagy telefonkábel
- a főkábel szegmens hossza: maximum 300 m, repeater-rel 1200 m
- Az állomások elméleti maximális száma: 255. A hatékonyan üzemeltethető állomások száma alkalmazásfüggő: a hálózatokba kapcsolt erőforrások és a felhasználás jellege együttesen határozzák meg.

SZOFTVER

— PROPOS V.3.30 vagy ezzel kompatíbilis operációs rendszer
— PRONET 3.0 hálózati szoftver
  - IBM PC NETWORK program kompatíbilis
  - PRONET BIOS (IBM NETBIOS EMULÁCIÓ)
  - FILE SERVER
  - PRINTER SPOOL SERVER
  - MESSAGE SERVER
  - PRONET—BASE a dBASE, CLIPPER hálózati kiegészítése
  - Elektronikus Posta
— A hálózat szolgáltatásainak elérése külső parancsok segítségével, operátori konzolról vagy programból (felhasználói interfész) biztosított

AZ SZKI — STABIL PARTNER!

Számítástechnikai Kutató Intézet és Innovációs Központ

1251 Budapest, Pf. 19.

Információ:

Sci-L **Számítástechnikai Informatikai Fejlesztő Leányvállalat**
1011 Budapest I., Iskola u. 10.

SCITEL Számítástechnikai Fejlesztő Leasing Leányvállalat
1015 Budapest I., Donáti u. 35—45.